VAV Systems

User Guide

DISTECH CONTROLS™

Legal

While all efforts have been made to verify the accuracy of information in this manual, Distech Controls is not responsible for damages or claims arising from the use of this manual. Persons using this manual are assumed to be trained HVAC professionals and are responsible for using the correct wiring procedures, correct override methods for equipment control and maintaining safe working conditions in fail-safe environments. Distech Controls reserves the right to change, delete or add to the information in this manual at any time without notice.

Distech Controls, the Distech Controls logo, ECO-Vue, and Allure are trademarks of Distech Controls Inc.; BACnet is a registered trademark of ASHRAE; Niagara Framework and Niagara^{AX} Framework is a registered trademark of Tridium, Inc.

Table of Contents

AirFlowControl block

AirFlow Control block (ArFlCt)

The *AirFlowControl (ArFlCt)* block calculates an active airflow setpoint used to operate the connected VAV unit.

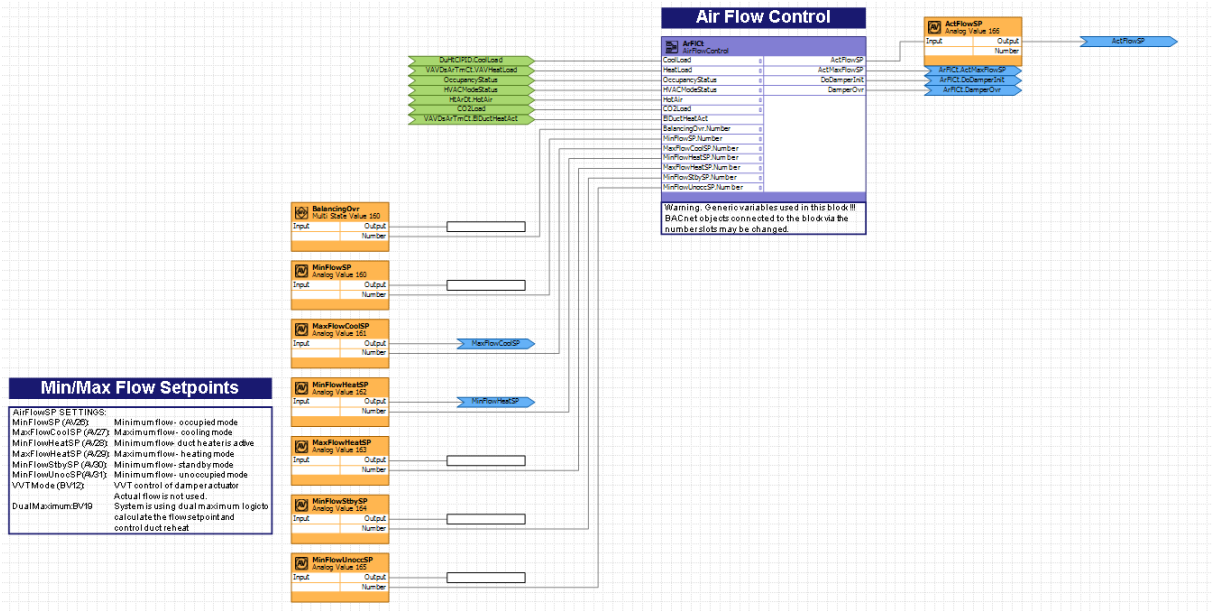


Figure 1: The AirFlowControl (ArFlCt) programming block with corresponding variables.

Inputs

The *AirFlowControl (ArFlCt)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
HeatLoad	A heating demand signal. (AV)
CoolLoad	A cooling demand signal. (AV)
OccupancyStatus	The Occupancy Status signal. (MV)
HVACModeStatus	An HVAC mode status carrying information on the current mode of the unit. (MV)
HotAir	A confirmation signal that the air temperature supplied from an AHU to the VAV's intake is high enough to use it for heating. (BV)
CO2Load	A CO2 control demand signal. (AV)
BalancingOvr.Number	A balancing mode override command. (MV reference) WARNING. The block uses this point to modify the connected variable.
MinFlowSP.Number	Minimum airflow setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
MaxFlowCoolSP.Number	Maximum cooling mode airflow setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
MinFlowHeatSP.Number	Minimum heating mode airflow setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
MaxFlowHeatSP.Number	Maximum heating mode airflow setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.

Input Parameter	Description
MinFlowStbySP.Number	Minimum standby mode airflow setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
MinFlowUnoccSP.Number	Minimum standby mode airflow setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.

Optional Inputs

Input Parameter	Description
ElDuctHeatAct	A request to force a minimum airflow due to an electrical duct heater being used. Default *False* (BV)

Outputs

The *AirFlowControl (ArFlCt)* block output signals (hardware and logic).

Output Parameter	Description
ActFlowSP	Active airflow setpoint (AV)
ActMaxFlowSP	Active maximum airflow setpoint status signal (AV)
DoDamperInit	Damper initialization command (BV)
DoDamperInit	Damper override command (AV)

Default BACnet objects

The *AirFlowControl (ArFlCt)* block related BACnet objects.

Output Parameter	Description
BalancingOvr (MV 160)	A balancing mode override command. BACnet Multistate Value 160 input/output signal (can be modified inside of the block through the variable reference number).
MinFlowSP (AV 160)	Minimum airflow setpoint. BACnet Analogue Value 160 input/output signal (can be modified inside of the block through the variable reference number).
MaxFlowCoolSP (AV 161)	Maximum cooling mode airflow setpoint. BACnet Analogue Value 161 input/output signal (can be modified inside of the block through the variable reference number).
MinFlowHeatSP (AV 162)	Minimum heating mode airflow setpoint. BACnet Analogue Value 162 input/output signal (can be modified inside of the block through the variable reference number).
MaxFlowHeatSP (AV 163)	Maximum heating mode airflow setpoint. BACnet Analogue Value 163 input/output signal (can be modified inside of the block through the variable reference number).
MinFlowStbySP (AV 164)	Minimum standby mode airflow setpoint. BACnet Analogue Value 164 input/output signal (can be modified inside of the block through the variable reference number).
MinFlowUnoccSP (AV 165)	Minimum standby mode airflow setpoint. BACnet Analogue Value 165 input/output signal (can be modified inside of the block through the variable reference number).
ActFowSP (AV 166)	Active airflow setpoint. BACnet Analogue Value 166 output signal.

Block Functions

OccupancyStatus algorithm

The block validates that the airflow setpoints do not conflict with one-another and modifies them if need be. Then the block checks whether air supplied from a superordinate AHU is adequate for heating or cooling mode and includes the CO2Load signal into the calculation.

If the BalancingOvr signal is active, it sets the DamperOvr and DoDamperInit outputs and overrides the airflow setpoint value calculated in the previous algorithm stages.

AirFlow setpoints validation

The block ensures that:

- the MinFlowSP ≥ 0;
- the MinFlowHeatSP ≥ 0;
- the MaxFlowCoolSP ≥ the MinFlowSP;
- the MaxFlowHeatSP ≥ the MinFlowHeatSP;
- the MaxFlowStbySP ≥ the MinFlowSP;
- the MaxFlowUnoccSP ≥ the MinFlowSP.

If required, it uses the reference number connections to modify BACnet Analogue Values holding the conflicting setpoints.

OccupancyStatus based setpoint selection

The OccupancyStatus sets minimum airflow setpoint values:

- in the Occupied(1) and Bypass(3) modes the MinFlowSp is used;
- in the Unoccupied(2) mode the MinFlowUnoccSp is used;
- in the Standby(4) mode the MinFlowStbySp is used.

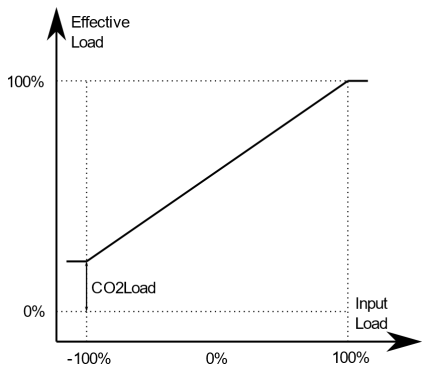


Figure 2: The CO2Load impact on the effective heating/cooling load.

CO2Load

The CO2Load signal changes the starting point of the Heat/CoolLoad characteristic to ensure increased airflow in case of high CO2 concentration.

Heating/Cooling action

The block checks a value of the HotAir input and:

- the HotAir is active -> the HeatLoad drives the ActFlowSP between the MinFlowSP (or if ElDuctHeatAct is true, the MinFlowHeatSP) and the MaxHeatFlowSP ;
- the HotAir is not active -> the CoolLoad drives the ActFlowSP between the MinFlowSP and the MaxCoolFlowSP

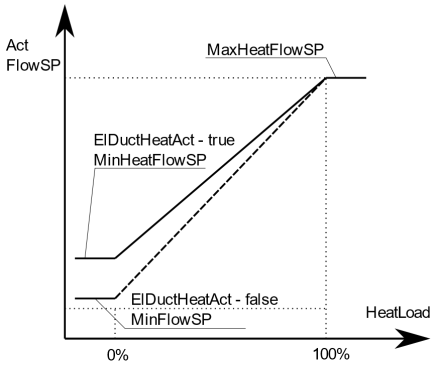


Figure 3: The ActFlowSP control in the heating mode.

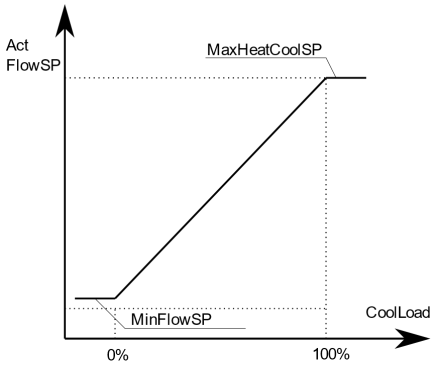


Figure 4: The ActFlowSP control in the cooling mode.

HVACModeStatus impact on the ActFlowSP

The heating/cooling load based airflow setpoint is subjected to a HVACModeStatus filter:

- if the HVACModeStatus is Off(7), the ActFlowSP is set to 0 L/s/cfm;
- if the HVACModeStatus is Heat(2), MrngWarmup(3), EmergHeat(9) or MaxHeat(13), the ActFlowSP follows the HeatLoad;
- if the HVACModeStatus is Cool(4), NightPurge(5), PreCool(6), FreeCool(11), Ice(12), the ActFlowSP follows the CoolLoad;
- if the HVACModeStatus is Auto(1), Test(8), FanOnly(10), Economy(14), Dehumid(15), the ActFlowSP switches between the CoolLoad and HeatLoad in the function of the HotAir input.

ActMaxFlowSP

The ActMaxFlowSP is set to the MaxFlowCooSP or the MaxFlowHeatSP following the HVACModeStatus:

- if the HVACModeStatus is Off(7), the ActMaxFlowSP is set to 1 L/s/cfm;
- if the HVACModeStatus is Heat(2), MrngWarmup(3), EmergHeat(9) or MaxHeat(13), the ActMaxFlowSP follows the MaxFlowHeatSP;
- if the HVACModeStatus is Cool(4), NightPurge(5), PreCool(6), FreeCool(11), Ice(12), the ActMaxFlowSP follows the MaxFlowCooSP;
- if the HVACModeStatus is Auto(1), Test(8), FanOnly(10), Economy(14), Dehumid(15), the ActMaxFlowSP switches between the MaxFlowCooSP and MaxFlowHeatSP in the function of the HotAir input.

BalancingOvr

The BalancingOvr can be used to commission the VAV. It uses a reference number connection to a BACnet Multistate object and can be set to:

1. Normal operation – default state, no override active;
2. GotoMinFlow – sets the ActFlowSP to the MinFlowSP;
3. GotoMinFlowHeat – sets the ActFlowSP to the MinFlowHeatSP;
4. GotoMaxFlowCool – sets the ActFlowSP to the MaxFlowCoolSP;
5. GotoMaxFlowHeat – sets the ActFlowSP to the MaxFlowHeatSP;
6. Goto75%MaxFlow – sets the ActFlowSP to 75% of the MaxFlowCoolSP;
7. Goto80%MaxFlow – sets the ActFlowSP to 80% of the MaxFlowCoolSP;
8. Goto85%MaxFlow – sets the ActFlowSP to 85% of the MaxFlowCoolSP;
9. Goto90%MaxFlow – sets the ActFlowSP to 90% of the MaxFlowCoolSP;
10. Goto95%MaxFlow – sets the ActFlowSP to 95% of the MaxFlowCoolSP;
11. FullOpen – forces the VAV’s damper actuator to a fully opened position;
12. FullClose – forces the VAV’s damper actuator to a fully closed position;
13. DoDamperInit – forces the VAV’s damper actuator to reinitialize;

If the BalancingOvr stays forced for more than 24 hours it will be reset to NormalOperation(1).

CO2LoadCalculation control block

CO2 Load Calculation block (CO2LdCa)

The *CO2LoadCalculation (CO2LdCa)* block uses a PID algorithm to control CO2 concentration level in an air-conditioned room.

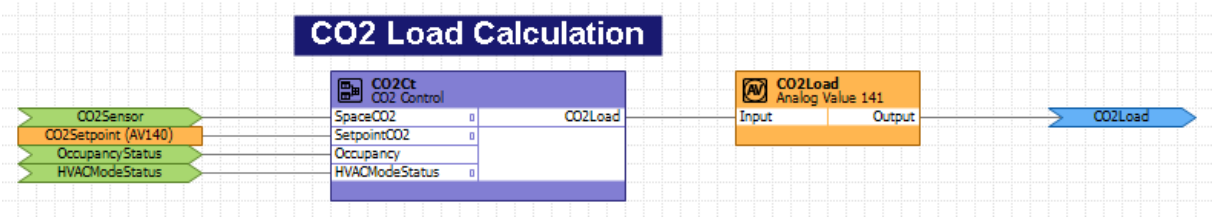


Figure 5: The CO2LoadCalculation (CO2LdCa) programming block with corresponding variables.

Inputs

The *CO2LoadCalculation (CO2LdCa)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
CO2Sensor	A CO2 condensation measurement. (AV)
CO2Setpoint	A CO2 condensation setpoint. (AV)
HVACModeStatus	An HVAC mode status input carrying information on the current mode of the unit. (MV)
OccupancyStatus	The Occupancy status signal carrying information on the current occupancy state of the room. (MV)

Outputs

The *CO2LoadCalculation (CO2LdCa)* block output signals (hardware and logic).

Output Parameter	Description
CO2Load	A CO2 control demand signal. (AV) (0%-100%)

Default BACnet objects

The *CO2LoadCalculation (CO2LdCa)* block related BACnet objects.

Output Parameter	Description
CO2Setpoint (AV140)	A CO2 condensation setpoint. BACnet Analogue Value 140 input signal.
CO2Load (AV141)	A CO2 control demand signal. BACnet Analogue Value 141 output signal. (0%-100%)

Block Functions

Operation principle

When the room is occupied (the OccupancyStatus is set to Occupied or Bypass mode), and the HVACModeStatus is not Off(7) or EmergHeat(9), the block will use a BACnet PID Loop to calculate CO2Load signal, which is used to control the volume of fresh air supplied to the room.

CO2 control

The CO2Load calculation is active when all three below mentioned conditions are fulfilled:

- the HVACModeStatus is not Off(7) nor EmergHeat(9);
- the OccupancyStatus is Occupied or Bypass, indicating a presence in the room;
- the CO2Sensor is in a valid range (between 0ppm and 5000ppm).

When activated the BACnet PID Loop takes in the CO2Sensor and CO2Setpoint and to calculate CO2Load output value, which is used to increase/decrease fresh air supply to the air-conditioned space.

DamperPosition block

DamperPosition control block (DmPo)

The *DamperPosition (DmPo)* block operates a VAV damper using an active airflow setpoint and overrides.

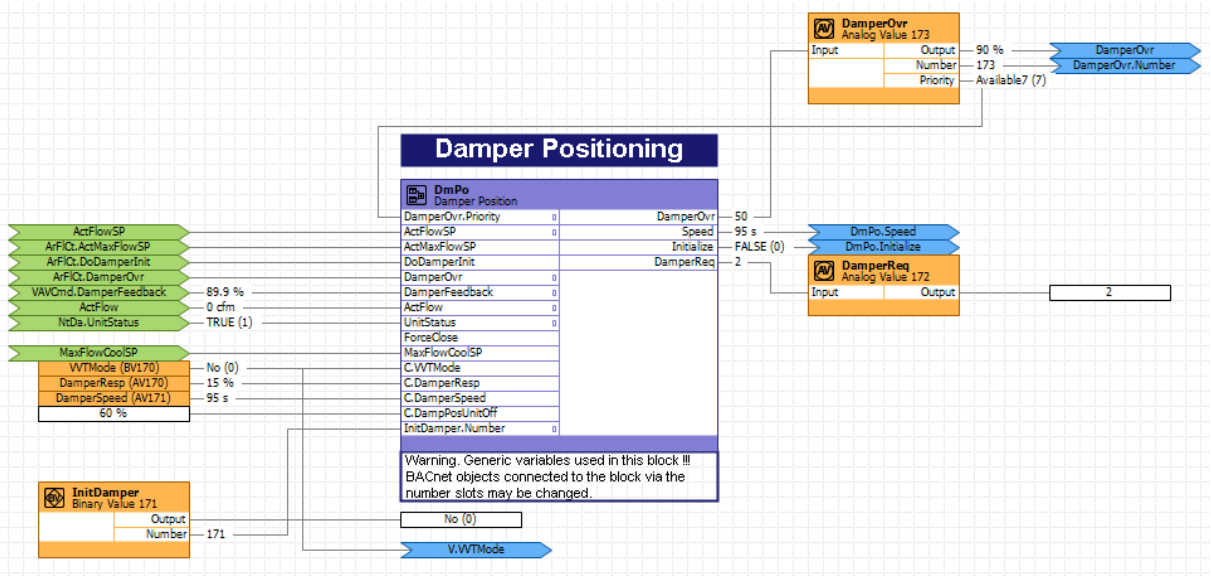


Figure 6: The DamperPosition (DmPo) programming block with corresponding variables.

Inputs

The *DamperPosition (DmPo)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
DamperFeedback	Damper position feedback. (AV)
ActFlow	A volumetric airflow measurement. (AV)
ActFlowSP	Active airflow setpoint. (AV)
UnitStatus	A superordinate AHU operational status signal. (BV)
InitDamper.Number	A damper initialization variable. (BV reference) WARNING. The block uses this point to modify the connected variable.
DamperOvr.Priority	Damper override variable BACnet priority value. (BACnet priority)
DamperOvr	Damper override signal from an airflow control block. (AV)

Optional inputs

Input Parameter	Description
BalancingMode	An interface signal uses with an EC-SmartVue sensor to activate balancing mode. Default *False* (BV)
ForceClose	An interface signal forcing damper closure when VAV's fan starts. Default *False* (BV)
ActMaxFlowSP	Active maximum airflow setpoint status signal. Default 450L/s/950cfm(AV)
MaxFlowCoolSP	Maximum cooling mode airflow setpoint. Default 250L/s/500cfm(AV)

Input Parameter	Description
DoDamperInit	Damper initialization active (a balancing mode interface signal). Default *False* (BV)

Configuration Parameters

Input Parameter	Description
C.VVTMode	Variable Volume & Temperature mode active. Default *False* (BV)
C.DamperResp	A multiplier (in %) applied to the calculated damper movement. It is used to adjust the reaction speed of the damper. Default *15%* (AV)
C.DamperSpeed	Damper full cycle time. Default 95*s* (AV)
C.DampPosUnitOff	Damper position when the unit is off. Default: 0%. (AV)

Outputs

The *DamperPosition (DmPo)* block output signals (hardware and logic).

Output Parameter	Description
DamperOvr	Damper override value. (AV)
Speed	Active maximum airflow setpoint status signal. (MV)
Initialize	Damper initialization command (BV)
DamperReq	Damper request coded signal. (AV)

Default BACnet objects

The *DamperPosition (DmPo)* block related BACnet objects.

Output Parameter	Description
VVTMode (BV 170)	Variable Volume & Temperature mode active. BACnet Binary Value 170 input signal.
DamperResp (AV 170)	A multiplier (in %) applied to the calculated damper movement. It is used to adjust the reaction speed of the damper. BACnet Analogue Value 170 input signal.
DamperSpeed (AV 171)	A full-cycle damper time. BACnet Analogue Value 171 input signal.
InitDamper (BV 171)	An interface signal uses with an EC-Smart-Vue sensor to reinitialize the damper actuator. BACnet Binary Value 171 input signal.
DamperOvr (AV 173)	Damper override value. BACnet Analogue Value 173 input/output signal.
DamperReq (AV 172)	Damper request coded signal. BACnet Analogue Value 172 output signal.

Block Functions

Damper Positioning algorithm

The block determines whether the damper should follow the calculated AirFlowSP, operate in the variable Volume & Temperature mode or accept overrides.

The damper might be forced by the following signals (listed from the highest to the lowest priority):

the ForceClosed input - if active, the damper is forced to 0%. Usually this signal is used in conjunction with the VAV fan block to ensure fan start with a correct rotation direction;

the UnitStatus if not active, indicating that a superordinate AHU does not operate, the damper output is set to the C.DampPosUnitOff parameter;

the C.VVTMode parameter – if active, the block recalculates the ActFlowSP from flow units to per cent using the ActMaxFlowSP as 100% reference point.

the DamperOvr.Priority is not equal to ProgramIP(16)/ProgramMSTP(14), indicating an external override of the damper output – the damper output follows the DamperFeedback to synchronize the ActuatorControl position with the actual damper position.

When none of the above-mentioned overrides is active, the damper is controlled in the VAV mode.

AirFlow control

The default operation mode is VAV when the controller follows the AirFlowSP and operates the damper to match it with the measured AirFlow value.

The block uses a built-in VAV ActuatorControl toolbox object to generate the DamperCmd output signal. The C.DamperResp parameter is used to minimize “damper hunting” due to the airflow overshooting the setpoint, and the MaxFlowCoolSP to optimize the next damper movement not to exceed the maximum flow value.

Balancing mode

The balancing mode might either be initiated by a BalancingOvr BACnet Multistate Value, used as an interface point for *my*DC AirBalancing mobile application, or an EC-Smart-Vue room sensor.

The BalancingOvr variable’s operation (for more details, refer to the AirFlowControl block section) influences the block via the DamperOvr.Priority, DamperOvr and DoDamperInit inputs. If the EC-Smart-Vue is used, it forces the DamperOvr and the InitDamper BACnet variables directly. The block knows that the DamperOvr overridden through the DamperOvr.Priority input.

Damper initialization

When the BACnet Binary Value InitDamper is set by the EC-Smart-Vue sensor balancing mode or the DoDamperInit input is active, the Initialize output is activated. Simultaneously the InitDamper variable is reset using the number reference.

When the BalancingOvr variable is used, the AirFlowControl block sends the signal to DoDamperInit input to reinitialize the damper actuator.

DualHeatCoolPID control block

Dual Heat Cool PID control block (DuHtClPID)

The *DualHeatCoolPID (DuHtClPID)* block calculates heating and cooling load signals required to keep the space temperature at corresponding heating and cooling setpoints.

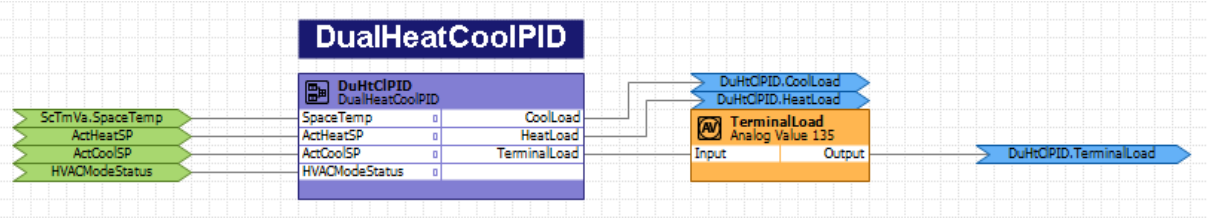


Figure 7: The DualHeatCoolPID (DuHtClPID) programming block with corresponding variables.

Inputs

The *DualHeatCoolPID (DuHtClPID)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
SpeceTemp	Effective measured Space Temperature value. (AV)
ActHeatSP	Active heating setpoint. (AV)
ActCoolSP	Active cooling setpoint. (AV)
HVACModeStatus	An HVAC mode status carrying information on the current mode of the unit. (MV)

Optional Inputs

Input Parameter	Description
DischAirHeatLimit	Current value of a Discharge Air Temp based heating action limitation. Default *100%* (AV)
DischAirCoolLimit	Current value of a Discharge Air Temp based cooling action limitation. Default *100%* (AV)
CondSensorSt	Condensation protection sensor. Default *False* (BV)

Outputs

The *DualHeatCoolPID (DuHtClPID)* block output signals (hardware and logic).

Output Parameter	Description
CoolLoad	A cooling demand signal. (AV) (0%-100%)
HeatLoad	A heating demand signal. (AV) (0%-100%)
TerminalLoad	A coded heating/cooling demand signal. (AV) (from -100% to 0% - heating, from 0% to 100% - cooling)

Default BACnet objects

The *DualHeatCoolPID (DuHtClPID)* block related BACnet objects.

Output Parameter	Description
TerminalLoad (AV135)	A coded heating/cooling demand signal. (from -100% to 0% - heating, from 0% to 100% - cooling) BACnet Analogue Value 135 output signal.

Block Functions

Heating and cooling PID activation

The block uses two separated BACnet PID Loop objects to independently control heating and cooling actions.

The heating action PID is activated when all the following conditions are fulfilled:

- The cooling PID is not active;
- The HVACModeStatus is in one of the following states Auto(1), Heat(2), MrngWrmUp(3), Test(8), EmergHeat(9), MaxHeat(13), Economy (14) or Dehumid(15).

The cooling action is activated when all the following conditions are fulfilled:

- The heating PID is not active;
- The CondSensorSt is not active and the HVACModeStatus is in one of the following states Auto(1), Cool(4), NightPurge(5), PreCool(6), Test(8), Ice(12), Economy(14) or Dehumid(15);
- Optionally if the CondSensorSt is active, but the HVACModeStatus is in the FreeCool(11) mode.

Discharge Air Temp activation

The block takes DischAirCoolLimit and DischAirHeatLimit signals to follow limitations imposed by a DischAirTemp controller. Heating and cooling load signals will be recalculated in accordance with the following graph.

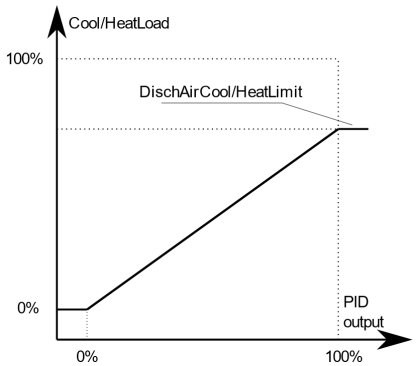


Figure 8: DischAirCool/HeatLimit impact on the Cool/HeatLoad signal.

Condensation Sensor

When the the CondSensorSt is active the cooling PID's action will be blocked, except when the HVACModeStatus is in the FreeCool(11) mode.

Terminal Load

The TerminalLoad is a coded signal, which combines information on the heating and cooling load in one data point. When the heating load is active, the TerminalLoad follows it but accepts negative values (0% -> 0% and 100% -> -100%). And when the cooling load is active, ,the TreminalLoad follows it directly (0% -> 0% and 100% -> -100%).

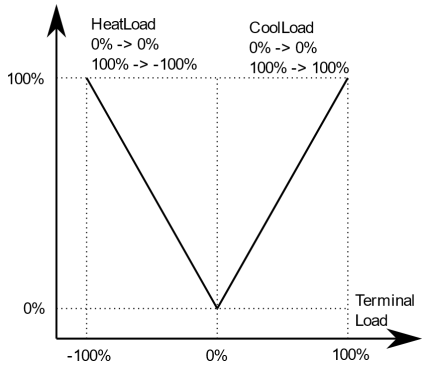


Figure 9: DischAirCool/HeatLimit impact on the Cool/HeatLoad signal.

EC-Smart-Vue configuration VAV

EC-Smart-Vue sensor sequence for VAV units

The *EC-Smart-Vue* sequence for VAV consists of multiple blocks combined in one snippet.

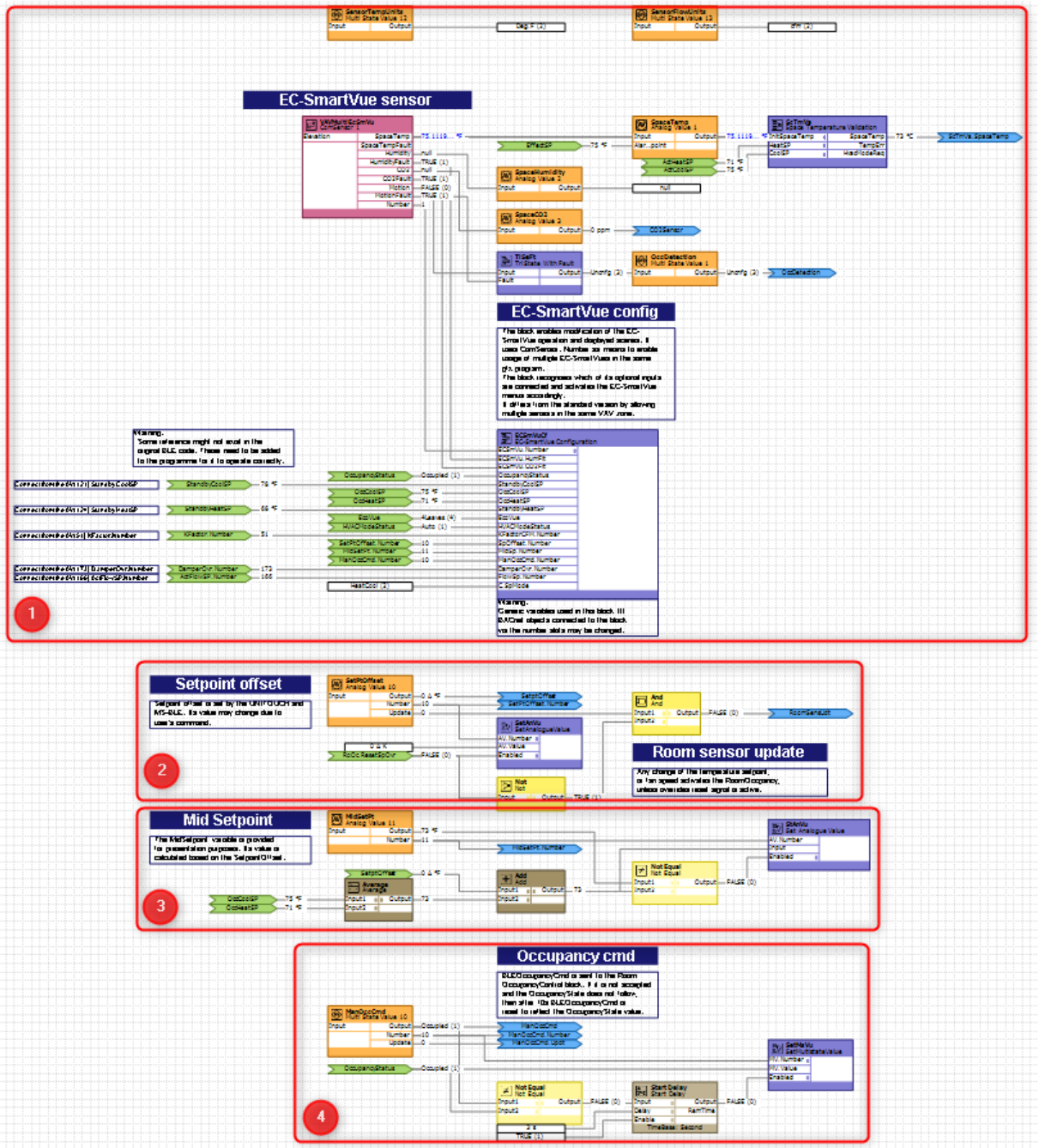


Figure 10: The EC-SmartVue programming sequence.

Snippet Structure

The *EC-Smart-Vue* snippet uses the same set of variables as the standard UNITOUCH based code. This approach unifies interactions between the main code and room sensor section. The snippet consists of four major sections:

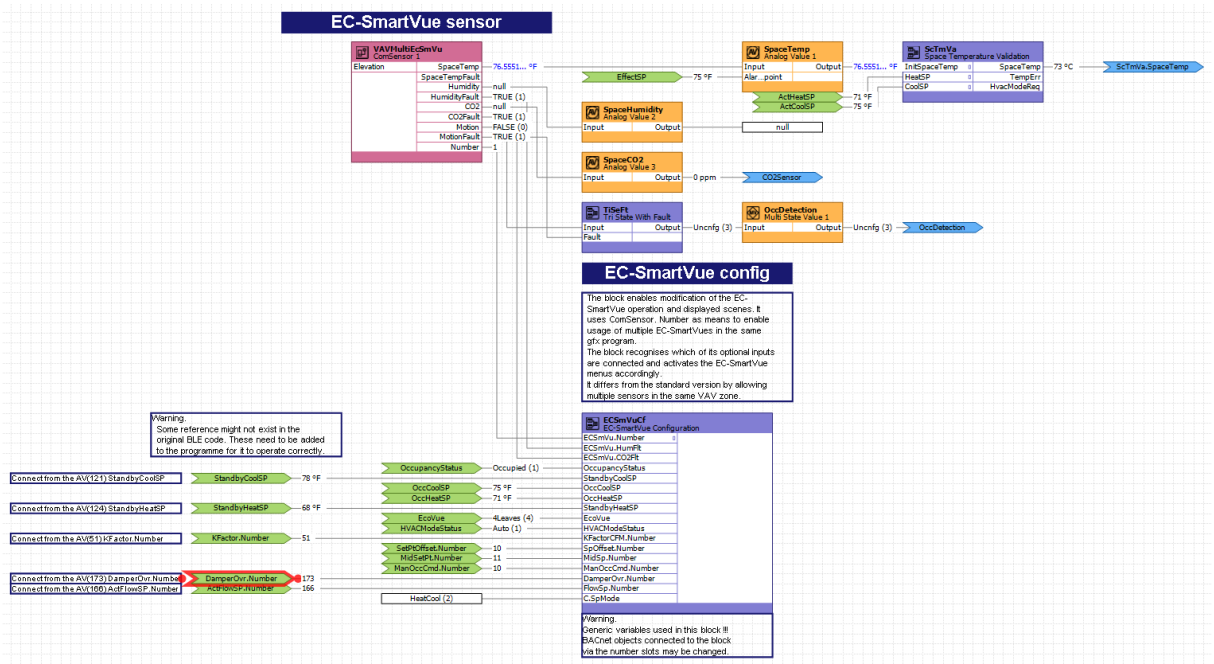
1. *ComSensor* block (a code representation of a hardware *EC-Smart-Vue* sensor) with *EC-Smart-Vue Configuration* section.
2. *SetPtOffset* update section.
3. *MidSetPt* update section.
4. *ManOccCmd* update section.

Sections 2-3 are mirror images of similar sections found in the standard application using the UNITOUCH sensor. They provide a unified set of variables which on one hand create an interface for user interaction via a room sensor or a BMS system, and on the other hand, send information to the VAV control code.

Section 1 creates an interface between the *EC-Smart-Vue* room sensor's intricate menu options and the variables set in sections 2-4.

EC-Smart-Vue Configuration block

The *EC-Smart-Vue Configuration* block provides the necessary interface between the ComSensor block and the VAV code.



The *EC-Smart-Vue Configuration* and the *ComSensor* blocks.

Inputs

The *EC-Smart-Vue Configuration* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
ECSmVu.Number	The ComSensor number. (AV)

Optional Inputs

Input Parameter	Description
ECSmVu.HumFlt	EC-Smart-Vue humidity sensor fault. Default *true* (BV)
ECSmVu.CO2Flt	EC-Smart-Vue CO2 sensor fault. Default *true* (BV)
OccupancyStatus	The unit's effective OccupancyStatus. Default *Occupied(1)* (MV)
StandbyCoolSP	A Standby mode cooling setpoint. Default *28ºC, 78ºF* (AV)
OccCoolSP	An Occupied mode cooling setpoint. Default *23ºC, 75ºF* (AV)
OccHeatSP	An Occupied mode heating setpoint. Default *21ºC, 71ºF* (AV)
StandbyHeatSP	A Standby mode cooling setpoint. Default *18ºC, 68ºF* (AV)
EcoVue	An energy efficiency indicator. Default *4leaves* (AV)
HVACModeStatus	The effective HVACModeStatus. Default *Auto(1)* (MV)
KFactor.Number	The KFactor Analogue Variable reference number. Default *null* (AV)
SpOffset.Number	The SpOffset Analogue Variable reference number. Default *null* (AV)
MidSp.Number	The MidSp Analogue Variable reference number. Default *null* (AV)
ManOccCmd.Number	The ManOccCmd Multistate Variable reference number. Default *null* (AV)
DamperOvr.Number	The DamperOvr Analogue Variable reference number. Default *null* (AV)
FanSpeedCmd.Number	The FanSpeedCmd Multistate Variable reference number. Default *null* (AV)
FlowSp.Number	The FlowSp Analogue Variable reference number. Default *null* (AV)

Configuration Inputs

Input Parameter	Description
C.SpMode	The setpoint presentation mode. Default *SpOffset(1)* (AV)
C.EnableBalancing	Allow EC-Smart-Vue balancing mode. Default *true* (BV)
C.DispEffTempSp	Display the effective temperature setpoint. Default *true* (BV)

Default BACnet objects

The *EC-Smart-Vue Configuration* block and the *EC-Smart-Vue* sensor itself use BACnet objects. In two possible ways. Some of them are connected directly via a reference in the ComSensor interface menu, while others are connected via the reference numbers to the corresponding BACnet objects.

EC-Smart-Vue Configuration block connected variables

These variables are connected through reference number

BACnet object	Description
KFactor.Number	The KFactor Analogue Variable reference number. Default *null* (AV)
SpOffset.Number	The SpOffset Analogue Variable reference number. Default *null* (AV)
MidSp.Number	The MidSp Analogue Variable reference number. Default *null* (AV)
ManOccCmd.Number	The ManOccCmd Multistate Variable reference number. Default *null* (AV)
DamperOvr.Number	The DamperOvr Analogue Variable reference number. Default *null* (AV)
FanSpeedCmd.Number	The FanSpeedCmd Multistate Variable reference number. Default *null* (AV)
FlowSp.Number	The FlowSp Analogue Variable reference number. Default *null* (AV)

All these variables, except for the FanSpeedCmd (which is by default left unused in the code), are connected to both ComSensor and the EC-Smart-Vue Configuration block.

ComSensor connected variables

These variables are connected through the ComSensor configuration window.

BACnet object	ComSensor variable	Description
SensorFlowUnits (MV12)	IconFlowUnits	A value setting the display units for airflow values. BACnet Multistate Value.

BACnet object	ComSensor variable	Description
SensorTempUnits (MV13)	IconTempDeltaUnits IconTempUnits	A value setting the display units for temperature values. BACnet Multistate Value.
EffectSP (AV128)	NumEffSp	An effective temperature setpoint. BACnet Analogue Value.
MidSetPt (AV11)	NumMidSp	An effective midpoint setpoint. BACnet Analogue Value.
OccCoolSP (AV122)	NumOccCoolSP	An occupied cooling setpoint. BACnet Analogue Value
OccHeatSP (AV123)	NumOccHeatSP	An occupied heating setpoint. BACnet Analogue Value
SetPtOffset (AV10)	NumSpOffset	An effective offset value. BACnet Analogue Value
DamperDir (BV200)	EnumDamperDir	A damper direction definition. BACnet Binary Value
ActFlow (AV50)	NumActFlow	An effective air flow measurement. BACnet Analogue Value
ActFlowSP (AV166)	NumActFlowSP	An air flow setpoint. BACnet Analogue Value
DamperOvr (AV171)	NumDamperOvr	A damper position override. BACnet Analogue Value
VAV Actuator Position (AI106)	NumDamperPos	A damper position. BACnet Analogue Input
Differential Pressure (AI105)	NumDiffPress	A differential pressure value. BACnet Analogue Input
FlowCalib (AV52)	NumFlowCalib	Air flow calibration value. BACnet Analogue Value
InitDamper (BV171)	NumInitDamper	Damper initialization command. BACnet Binary Value
MaxFlowHeatSP (AV160)	NumMaxFlowHeatSP	Minimum airflow setpoint. BACnet Analogue Value
MaxFlowCoolSP (AV161)	NumMaxFlowSP	Maximum cooling mode airflow setpoint. BACnet Analogue Value
MinFlowHeatSP (AV162)	NumMinFlowHeatSP	Minimum heating mode airflow setpoint. BACnet Analogue Value
MinFlowSP (AV163)	NumMinFlowSP	Maximum heating mode airflow setpoint. BACnet Analogue Value
MinFlowStbySP (AV164)	NumMinFlowStbySP	Minimum standby mode airflow setpoint. BACnet Analogue Value
MinFlowUnoccSP (AV165)	NumMinFlowUnoccSP	Minimum standby mode airflow setpoint. BACnet Analogue Value

Block Functions

Display scenes activation

The *EC-Smart-Vue Configuration* block checks the E.ECSmVu.HumFlt and the E.ECSmVu.CO2Flt to determine, whether the humidity and CO2 measurements are valid and accordingly enables display scenes of the EC-Smart-Vue interface. Similarly, it checks the C.DispEffTempSp parameter to decide if the effective temperature setpoint scene should be shown.

Setpoint edit mode

The C.SpMode parameter decides which of the three setpoint modification methods will be used:

1. *SpOffset* – the relative setpoint offset method is used. User will be able to shit the middle control point by +/- 3ºC or 5ºF. In the background, the setpoint offset value is recalculated and the middle point setpoint is updated accordingly, so they are interchangeable at any given time.
2. *HeatCool* – the user operates directly on the OccHeatSP and OccCoolSP values. When the HVACModeStatus is in the Auto(1), Cool(4), NightPurge(5), PreCool(6), Off(7), Test(8), FanOnly(10), FreeCool(11), Ice(12), Economy(14) and Dehumid(15) state, the quick edit menu will display the OccCoolSP, in the Heat(2), MrgnWrmup(3), EmergHeat(9) and MaxHeat(13) states the quick edit menu will display the OccHeatSP. The block will limit changes to a valid range set by the Min/Max Heat/Cool values and by a code representation relation between the heating and cooling setpoint.
3. *MidSp* – the absolute middle point setpoint method is used. User will be able to set the setpoint to a range of values +/- 3ºC or 5ºF around the average of the OccCoolSP and the OccHeatSP. In the background, the setpoint offset value calculated based on the middle point setpoint, so they are interchangeable at any given time.

Occupancy command

The block uses the connected OccupacyStatus as a source of the current mode. When a user modifies the occupancy command, the block sets the OccCmd BACnet variable to one of the four basic states Occupied(1), Unoccupied(2), Bypass(3) or Standby(4), then the main control sequence checks the command's validity and sets the OccupancyStatus accordingly.

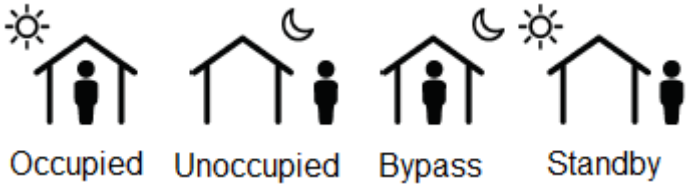


Figure 11: Occupancy icons.

ECO-Vue™ icon

The block uses the ECO-Vue input to show energy efficiency indicator in the form of a small twig with up to 4 leaves on it. The more leaves are shown on the twig the higher is the energy efficiency of the room comfort settings. This element is intended to show occupants how their choices impact room energy consumption and help raise eco-awareness.

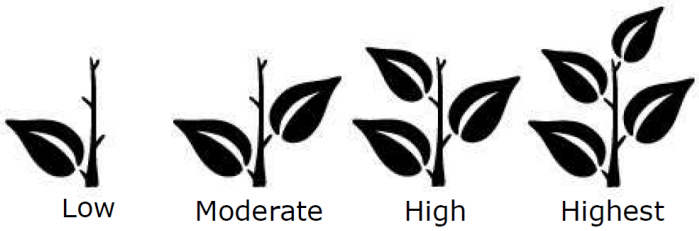


Figure 12: ECO-Vue energy efficiency levels.

HVACMode icon

The HVACModeStatus input enumeration needs to be translated to an EC-Smart-Vue specific HVAC icon enumeration. The block uses the following scheme:

HVAC icon	HVACModeStatus
Hidden(0)	Off(7)
Heat(1)	Heat(2), MrgnWrmup(3), MaxHeat(13)
Cool(2)	Cool(4), NightPurge(5), PreCool(6), FreeCool(11), Ice(12)
EmergencyHeat(3)	EmergHeat(9)
Dehumidify(8)	Dehumid(15)
Auto(10)	Auto(1), FanOnly(10), Economy(14)
Blink Auto Off(40)	Test(8)

Fan Speed command

The FanSpeedCmd number reference is used to show current fan speed using the EC-Smart-Vue icon and activate a command scene, which enables a user to cycle the FanSpeedCmd between Auto(1), Off(2), Low(3), Med(4) and High(5).

Balancing menus

The block used the C.EnableBalancing parameter to decide whether to activate VAV balancing menus. When active they allow:

- K Factor modification;
- Flow setpoint override;
- Flow measurement check and calibration;
- Damper position override;
- Pressure value check;
- Damper direction reversal;
- Damper reinitialization.

KFactor and Differential Pressure values will be recalculated depending on the IconFlowUnits variable. The damper and flow setpoint overrides will be written to at priority 7, which gives them precedence over standard manual overrides.

For additional information on the EC-Smart-Vue sensor, please refer to the Allure EC-Smart-Vue spec sheet, hardware installation guide, and user guide.

ECOVue block

ECO Vue Calculation block (ECVeCl)

The ECOVueCalculation (ECVeCl) block calculates the EcoVue value used as an energy efficiency indicator for space occupants.

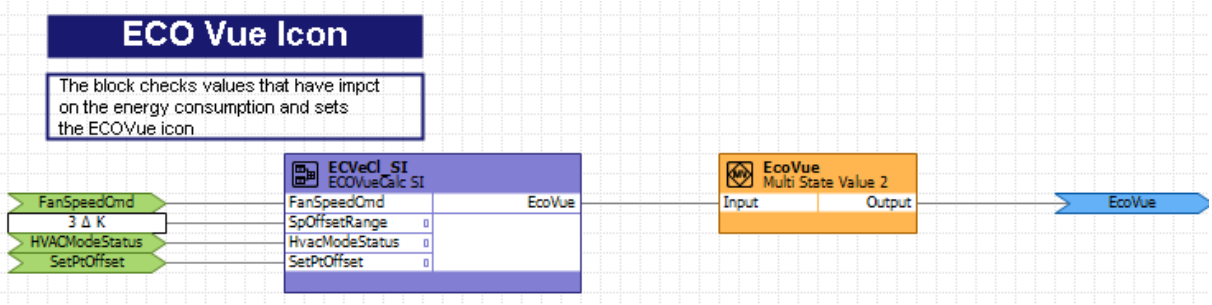


Figure 13: The ECOVueCalculation (ECVeCl) programming block with corresponding variables.

Inputs

The ECOVueCalculation (ECVeCl) block takes in the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
SpOffasetRange	A maximum range of the SetPtOffset variable. (AV)
HVACModeStatus	An HVAC mode status input carrying information on the current mode of the unit. (MV)
SetPtOffset	An active setpoint offset value. (AV)

Optional Inputs

Input Parameter	Description
FanSpeedCmd	An active user-generated fan-speed command. Default Off(1)(MV)

Outputs

The *ECOVueCalculation (ECVeCl)* block output signals (hardware and logic).

Output Parameter	Description
EcoVue	An energy efficiency indicator. (AV) (1-4 “leaves”)

Block Functions

ECO-Vue signal

Some of the Distech Controls’ room sensors can show an energy efficiency indicator called EcoVue. It has a form of a small twig with 1- 4 leaves. The more leaves are shown on the twig the higher is the energy efficiency of the room comfort settings. This element is intended to show to occupants how their choices impact room energy consumption and help raise eco-awareness.

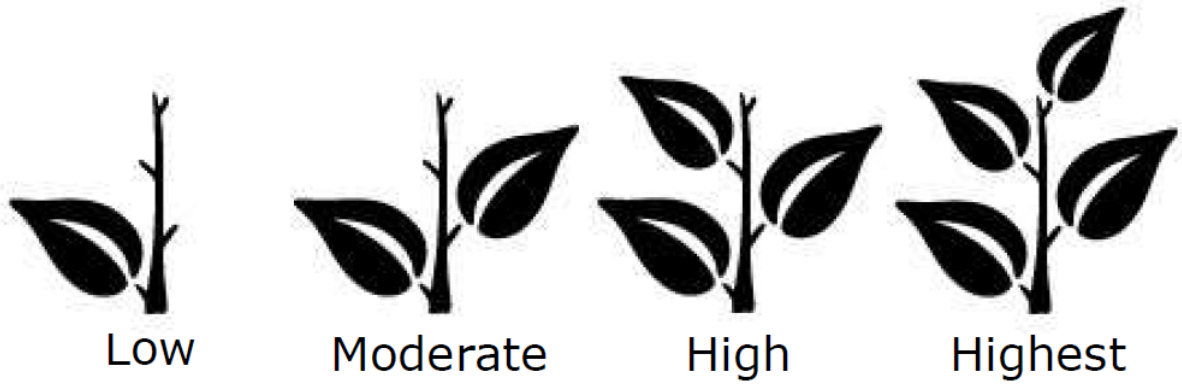


Figure 14: EcoVue energy efficiency levels

ECO-Vue calculation

The EcoVue output value is calculated based on two inputs – the SetPtOffset and the FanSpeedCmd.

The SetPtOffset is recalculated to a normalised range (from 1 to 4 leaves), using the SpOffasetRange as a reference range, and the HVACModeStatus to differentiate between the heating and the cooling mode.

Then the FanSpeedCmd's impact is taken into account. When it is in the Auto(2) mode, the EcoVue value is increased by 1 leaf, when it is forced to Low(3), Med(4) or High(5), the EcoVue is respectively decreased by 1, 2 or 3 leaves. When the FanSpeedCmd is Off(1) it has no impact on the EcoVue value.

ECO-Vue calculation example

Let's assume that:

- the SetPtOffset = 2ΔK
- the SpOffasetRange = 4ΔK
- the HVACModeStatus = Heat(2)
- the FanSpeedCmd = Low(3)

Then, given that the unit in the heating mode, the setpoint offset is normalized to:

$$\frac{\text{SetPtOffset}}{\text{SpOffasetRange}} \cdot 3 = \frac{2}{4} \cdot 3 \approx 1\ leaf \xrightarrow{HeatMode} 4 - (1\ leaf) = 3\ leaves$$

Next the FanSpeedCmd, equal to Low(3), decreases the number of leaves by 1, resulting in 2 leaves. The symbol will show:

Moderate

If all the same values were used but the HVACModeStatus would be Cool(4), the result would be:

$$\frac{\text{SetPtOffset}}{\text{SpOffasetRange}} \cdot 3 = \frac{2}{4} \cdot 3 \approx 1\ leaf \xrightarrow{CoolMode} 4 + (1\ leaf) = 5\ leaves$$

Next the FanSpeedCmd, equal to Low(3), decreases the number of leaves to 1, resulting in 4 leaves. The symbol would show:

Highest

ElectricalHeater control block

Electrical Heater Control block (ElHtCt)

The *ElectricalHeaterControl (ElHtCt)* block uses a HeatLoad signal from the main PID controller and uses it to operate an electrical heater.

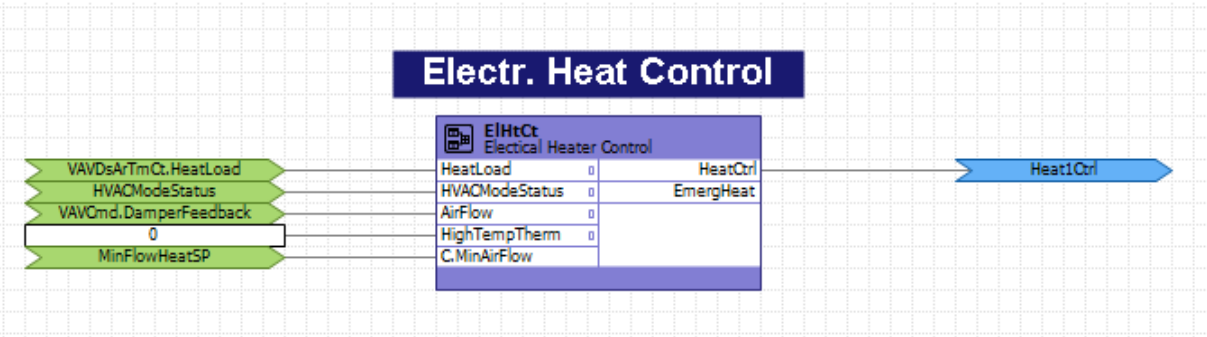


Figure 15: The ElectricalHeaterControl (ElHtCt) programming block with corresponding variables.

Inputs

The *ElectricalHeaterControl (ElHtCt)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
HeatLoad	A heating demand signal. (AV)
HVACModeStatus	An HVAC mode status input carrying information on the current mode of the unit. (MV)
AirFlow	An AirFlow signal (volumetric flow measurement AV/flow or pressure switch BV)
HighTemTerhm	The electrical heater overheat protection thermostat. (AV)

Configuration Parameters

Input Parameter	Description
C.RatioX1	A lower abscissa of the characteristic used to recalculate the HeatLoad input signal to HeatCtrl output. Default *0%*(AV)
C.RatioY1	A lower ordinate of the characteristic used to recalculate the HeatLoad input signal to HeatCtrl output. Default *0%*(AV)
C.RatioX2	A higher abscissa of the characteristic used to recalculate the HeatLoad input signal to HeatCtrl output. Default *0%*(AV)
C.RatioY2	A higher ordinate of the characteristic used to recalculate the HeatLoad input signal to HeatCtrl output. Default *0%*(AV)
C.MinAirFlow	A minimum airflow value required for the heater to operate without risk of overheating. Can be interpreted as an analogue or a binary value. Default 1(BV)

Outputs

The *ElectricalHeaterControl (ElHtCt)* block output signals (hardware and logic).

Output Parameter	Description
HeatCtrl	A heater control signal. (AV) (0%-100%)

Block Functions

Operation principle

When the heating mode is activated, the block checks whether the airflow is sufficient, and the high-temperature thermostat has not tripped and then uses the ratio block to recalculate the HeatLoad input signal to the HeatCtrl output. The latter is then used to operate the electrical heater directly.

Heating mode activation

The HeatCtrl is active when all three below mentioned conditions are fulfilled:

- the HVACModeStatus is in one of the following states Auto(1), Heat(2), MrngWrmUp(3), Test(8), EmergHeat(9), MaxHeat(13), Economy (14) or Dehumid(15);
- the AirFlow is sufficient for the electrical heater to operate;
- the HighTempTherm is not active.

Minimum AirFlow control

The block checks if the AirFlow input is greater than the C.MinAirFlow configuration parameter. A hysteresis block is used with a switch on point set to 95% of the C.MinAirFlow and a switch-off point set to 90% of the C.MinAirFlow.

The block can operate with both binary and analogue AirFlow signals.

If the AirFlow comes from a differential pressure or a flow switch in the form of a simple binary signal, the C.MinAirFlow needs to be set to value 1. Then when the AirFlow is active (equal to 1) the HeatCtrl follows the HeatLoad input, but when it is deactivated the HeatCtrl is locked to 0%.

If the AirFlow comes from an analogue flow sensor, the C.MinAirFlow must represent an actual flow value required for heater to operate without overheating. If the AirFlow is below the hysteresis points mentioned above, the HeatCtrl is locked to 0%.

High-Temperature Thermostat

When the HighTempTherm is active, the HeatCtrl is locked to 0%.

Ratio block calculation

The ratio block allows driving multiple heaters from the same HeatLoad signal. The abscissa values C.RatioX1 and C.RatioX2 enable assignment of any range of the HeatLoad to the full extent of the HeatCtrl output. The ordinate values C.RatioY1 and C.RatioY2 allow setting minimum and maximum value of the output signal.

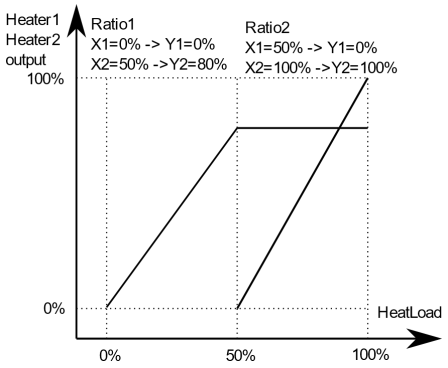


Figure 16: An example of using the ratio blocks to drive two heaters with one HeatLoad signal.

Emergency heat

When the HVACModeStatus is set to EmergHeat(9), the EmergHeat output is activated and if the AirFlow is present and the HighTempTherm has not tripped the HeatCtrl output is set to 100%.

HotAirDetection control block

Hot Air Detection control block (HtArDt)

The *HotAirDetection (HtArDt)* block checks whether the air temperature supplied from an AHU is sufficiently high for heating.

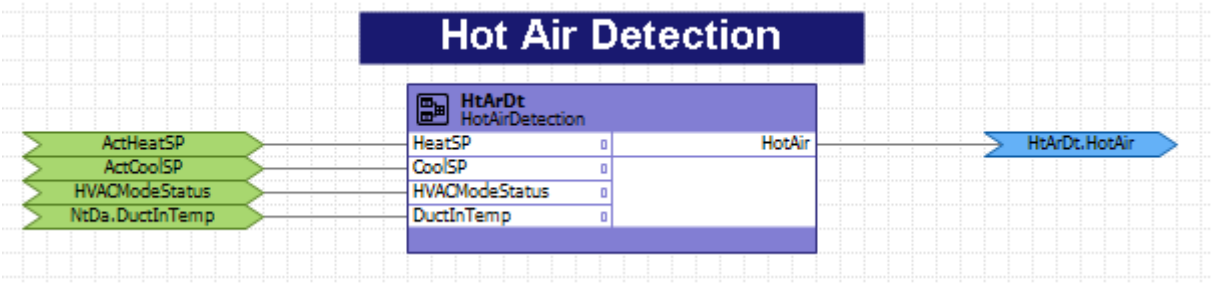


Figure 17: The HotAirDetection (HtArDt) programming block with corresponding variables.

Inputs

The *HotAirDetection (HtArDt)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
HeatSP	Active heating setpoint value. (AV)
CoolSP	Active cooling setpoint value. (AV)
HVACModeStatus	An HVAC mode status input carrying information on the current mode of the unit. (MV)
DuctInTemp	Air temperature supplied from an AHU. (AV)

Optional Inputs

Input Parameter	Description
ChgOverSwitch	A binary signal activated when the AHU is in the heating(1) and cooling(0) mode. (BV)

Outputs

The *HotAirDetection (HtArDt)* block output signals (hardware and logic).

Output Parameter	Description
HotAir	A status signal confirming that air temperature supplied from an AHU is sufficiently high for heating. (BV)

Block Functions

Operation principle

The block enables VAV's air heating mode, when one of the three conditions is fulfilled:

- The DuctInTemp is sufficiently high;

- The HVACModeStatus is set to Mrng_Wrmup(3);
- The ChgOvrSwitch input indicates that the superordinate AHU is in the heating mode.

DuctIntTemp check

The block uses a hysteresis algorithm to enable VAV's heating mode based on the DuctItTemp. The HotAir signal will be activated when the DuctInTmep is higher than the average value of the HeatSP and the CoolSP increased by 2K/3,5ºF. It will be deactivated when the DuctInTemp drops below the average value of the HeatSP and the CoolSP decreased by 1K/2ºF.

HVAC Mode VAV control block

HVAC Mode VAV control block (HVACMdVAV)

The *HVACModeVAV (HVACMdVAV)* block manages the operating mode of the unit.

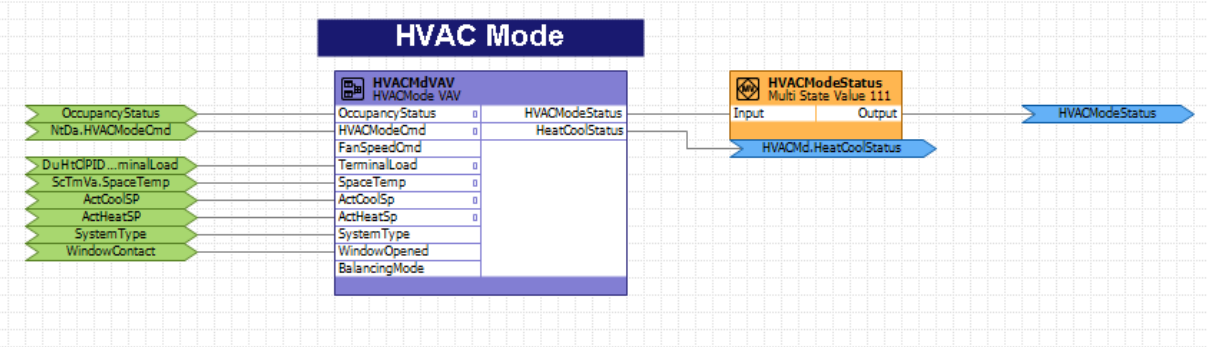


Figure 18: The HVACModeVAV (HVACMdVAV) programming block with corresponding variables.

Inputs

The *HVACModeVAV (HVACMdVAV)* block requires a connection to the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
OccupancyStatus	The Occupancy Status signal. (MV)
HVACModeCmd	An HVAC mode command usually coming from a network or a supervisory system. (MV)
TerminalLoad	A coded heating/cooling signal from the main PID controller. (AV) (from -100% to 0% - heating, from 0% to 100% - cooling)
SpaceTemp	Measured Space Temperature. (AV)
ActCoolSp	Active cooling setpoint. (AV)
ActHeatSp	Active heating setpoint. (AV)

Optional Inputs

Input Parameter	Description
FanSpeedCmd	Fan speed command usually from a room sensor/control panel. Default *Auto* (MV)
SystemType	An enumeration variable showing the system heating/cooling configuration. Default *CoolHeat* (MV)
WindowOpened	Current status of a window dry contact. Default *Unconfig* (MV)
BalancingMode	An interface variable used in conjunction with an EC-Smart-Vue room sensor. Default *False* (BV)
ChgOvr	Changeover status of two-pipe heating/cooling system. Default 0-*Cooling* (BV)

Configuration Parameters

Input Parameter	Description
C.ChgOverDelay	A switch delay between cooling and heating in changeover two-pipe systems. Default: 3*0s*(AV)

Outputs

The *HVACModeVAV (HVACMdVAV)* block output signals (hardware and logic).

Output Parameter	Description
HVACModeStatus	An effective HVAC mode status of the unit. (MV)
HeatCoolStatus	An effective heating/cooling status of the unit. (MV)

Default BACnet objects

The *HVACModeVAV (HVACMdVAV)* block related BACnet objects.

Output Parameter	Description
HVACModeStatus (MV111)	Holds information on current HVAC mode of the unit. From this variable, this piece of information is distributed to all other elements of the code. BACnet Multistate Value 111 output signal.

Block Functions

HVACModeStatus algorithm

The block checks input signals to determine the unit's HVACModeStatus, which is used in a different section of the code to invoke corresponding actions (for example activate heating/cooling PID or force Anti-Freeze action).

List of the all available HVACModes:

- AUTO Controller automatically determines an effective mode
- HEAT Heating only
- MRNG_WRMUP Application-specific morning warm-up
- COOL Cooling only
- NIGHT_PURGE Application-specific night purge
- PRE_COOL Application-specific pre-cool
- OFF Unit stopped, only safety actions active
- TEST Equipment being tested
- EMERG_HEAT Emergency heat mode (used for an anti-freeze action)
- FAN_ONLY Air not conditioned, the fan turned on
- FREE_COOL Cooling with cooling coil stopped
- ICE Ice-making mode
- MAX_HEAT Maximum heating mode
- ECONOMY Economic Heat/Cool mode
- DEHUMID Dehumidification mode

The block uses only modes Auto(1), Heat(2), Cool(4), Off(7) and EmergHeat(9). Other modes can be forced through the HVACModeCmd input.

OccupancyStatus and Startup reset

Each time the OccupancyStatus changes the HVACModeStatus is reset to OFF(7) for one program cycle to reinitialise control blocks in other sections of the application. Similar action is undertaken and upheld for 60s. after controller restart.

Effective HVACModeCmd

The effective HVACModeCmd follows the input value, except when:

- the WindowOpened signal indicates an opened window – the HVACModeCmd is overridden to OFF(7),
- the FanSpeedCmd signal is set to OFF(2) – the HVACModeCmd follows and is overridden to OFF(7),
- the original HVACModeCmd forces heating/cooling and the SystemType input indicates that corresponding hardware is not available - signal is set to OFF(7),
- the SpaceTemp drops below the C.AntiFreezeTemp – the HVACModeCmd is set to Emerg_Heat(9).

If the effective HVACModeCmd is set to Auto(1), the block determines heating/cooling mode in function of the SystemType, SpaceTemp, TerminalLoad, ActHeat/CoolSp and ChgOvr inputs.

Cooling mode activation

The cooling mode Cool(4) is activated when the SystemType and the ChgOvr indicate that cooling is possible and:

- the SpaceTemp rises above the ActCoolSp;
- or the TerminalLoad indicates a need for cooling action (its value is above 0%).

The cooling mode Cool(4) is deactivated when:

- the SystemType and the ChgOvr status indicate that cooling is not possible;
- or the SpaceTemp drops below the ActHeatSp;
- or the TerminalLoad is activated for the heating action (its value is below 0%).

Heating mode activation

The Heating mode Heat(1) is activated when the SystemType and ChgOvr indicate that heating is possible and:

- the SpaceTemp drops below the ActHeatSp;
- or the TerminalLoad indicates a need for heating action (its value is below 0%).

The Heating mode Heat(1) is deactivated when:

- the SystemType and the ChgOvr indicate that heating is not possible;
- or the SpaceTemp rises above the ActCoolSp;
- or the TerminalLoad is activated for the cooling action (its value is above 0%).

ChangeOver delay

The Heating and Cooling modes are mutually interlocked, meaning when one is enabled, the other is locked and cannot be switched on until the former is deactivated. To avoid going from cooling to heating and back in a rapid succession, a changeover delay is introduced. When any of the above-mentioned states deactivate, the other will go into action no earlier than after the C.ChgOvrDelay elapses (by default 30s).

Anti-freeze action

When the SpaceTemp drops below C.AntiFreezeTemp (default 8ºC, 46ºF) an EmergHeat(9) mode is activated. It takes precedence over all the other modes and is used by control block throughout the application to execute freeze protection precedures. This emergency mode is released when the SpaceTemp rises by a fixed hysteresis (default 2ΔºC, 3.5ΔºF).

Heat/Cool status

The HVACModeStatus is transferred to HeatCoolStatus output designed as an interface point to UNI-TOUCH and EC-Multi-Sensor BLE devices. The output will be set to:

1. Auto For HVAC modes Auto(1), Off(7), Test(8), FanOnly(10), Economy(14) and Dehumid(15);
2. Heat For HVAC modes Heat(2), MrngWrmUp(3), EmergHeat(9) and MaxHeat(13);
3. Cool For HVAC modes Cool(4), NightPurge(5), PreCool(6), FreeCool(11) and Ice(12).

BalancingMode action

When an EC-Smart-Vue sensor is used and a user activates the BalancingMode, the HVACModeStatus will be set to Auto(1) mode.

ParallelFanControl block

ParallelFanCtrl control block (PaFaCt)

The *ParallelFanCtrl (PaFaCt)* block operates a parallel VAV fan.

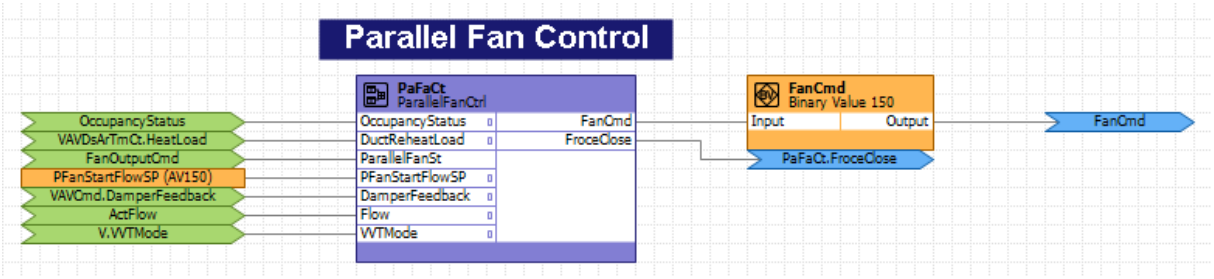


Figure 19: The ParallelFanCtrl (PaFaCt) programming block with corresponding variables.

Inputs

The *ParallelFanCtrl (PaFaCt)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
OccupancyStatus	Occupancy Status signal. (MV)
DuctReheatLoad	Reheater load control signal. Usually, the signal comes from a VAVDischargeAirTemp control block. (AV)
PFanStartFlowSP	Parallel fan start flow setpoint. (AV)
DamperFeedback	A feedback signal from the damper actuator hardware output block. (AV)
Flow	A volumetric airflow measurement. (AV)
VVTMode	Variable Volume & Temperature mode state. (BV)

Optional Inputs

Input Parameter	Description
ParallelFanSt	Fan operation feedback signal. The signal usually comes from the hardware output feedback output. Default *null* (BV)

Outputs

The *ParallelFanCtrl (PaFaCt)* block output signals (hardware and logic).

Output Parameter	Description
FanCmd	Fan start signal. (BV)
ForceClose	Damper closed override command. Should be connected to a ForceClose input of a Damper Control block. (BV)

Default BACnet objects

The *ParallelFanCtrl (PaFaCt)* block related BACnet objects.

Output Parameter	Description
PFanStartFlowSP (AV 150)	Parallel fan start flow setpoint value. BACnet Analogue Value 150 input signal.

Output Parameter	Description
FanCmd (BV 150)	Fan start command signal. BACnet Binary Value 150 output signal.

Block Functions

Parallel fan start algorithm

The parallel fan is started if any of the below mentioned occurs:

- the DuctReheatLoad is higher than 10% (a 10% switch-off hysteresis applies);
- the VAV is not in the Variable Volume & Temperature mode (VVTMode input not active), and the VAV is in the Occupied or Bypass OccupancyStatus, and the Flow input is below the PFanStartFlowSP (a 5L/s/cfm switch-off hysteresis applies);

When the start conditions are fulfilled, the block sends ForceClose signal, which needs to be connected to a ForceClose input of a DamperPosition control block, to force closing of the VAV damper. It is done to prevent reversed rotation of the fan by the airflow coming from the VAV. Once the ParallelFanCtrl block sees that DamperFeedback input is below 1% it activates FanCmd output to start the fan. Minimum on(120s) and off(120s) time block prevents the fan to be switched on/off too frequently.

Room Occupancy control block

Room Occupancy control block (RoOc)

The *RoomOccupancy (RoOc)* block manages the room occupancy and occupancy status of the unit.

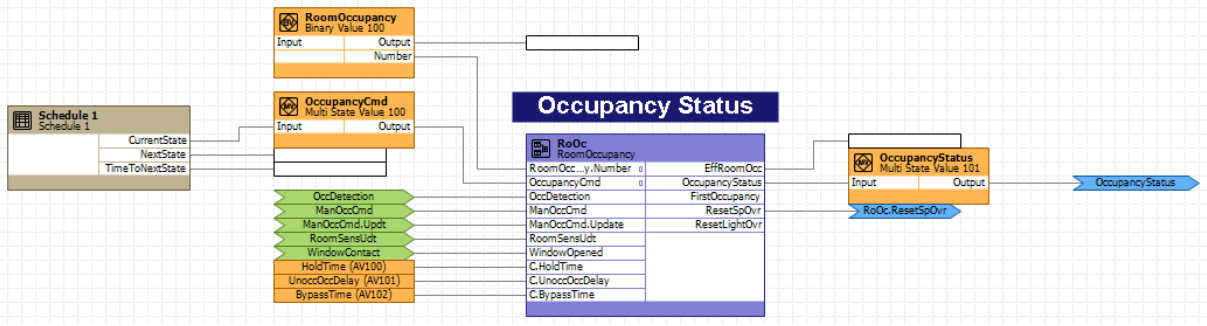


Figure 20: The RoomOccupancy (RoOc) programming block with corresponding variables.

Inputs

The *RoomOccupancy (RoOc)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
RoomOccupancy.Number	The Room Occupancy variable reference number. (BV reference) WARNING. The block uses this point to modify the connected variable.
OccupancyCmd	An Occupancy Command signal, usually coming from the schedule block. (MV)

Optional Inputs

Input Parameter	Description
OccDetection	Current status from a motion detection sensor. Default *Unconfig* (MV)
ManOccCmd	An occupancy command received from the network or through an operator interaction. Default *Occupied* (MV)
ManOccCmd.Update	An update of the ManOccCmd signal. Default *False* (BV)
RoomSensUdt	A signal indicating a user's interaction with the room sensor's interface. Default *False* (BV)
WindowOpened	Current status of a window dry contact. Default *Unconfig* (MV)

Configuration Parameters

Input Parameter	Description
C.HoldTime	Motion detection Holdup Time. Default: *900s*(AV)
C.UnoccOccDelay	Transition delay from the Unoccupied to Occupied Status, forced by the Room Occupancy detection. Default *600s*(MV)
C.BypassTime	Bypass mode duration time. Default *120m* (AV)
C.MotionLockDelay	Motion lock delay after the Room Occupancy changes. Default *10s* (AV)
C.EnableRoomUnocc	Configuration parameter enabling the OccupancyCmd to set the Room Occupancy to the Unoccupied state. Default *false* (BV)
C.OccTranslation	The default value of sun blinds translation value in the Occupied mode. Default *0%* (AV)

Input Parameter	Description
C.OccRotation	The default value of sun blinds rotation value in the Occupied mode. Default *90º* (AV)
C.UnoccTranslation	The default value of sun blinds translation value in the Unoccupied mode. Default *100%* (AV)
C.UnoccRotation	The default value of sun blinds rotation value in the Unoccupied mode. Default *0º* (AV)

Outputs

The *RoomOccupancy (RoOc)* block output signals (hardware and logic).

Output Parameter	Description
EffRoomOcc	The Room Occupancy set by the block. (BV)
OccupancyStatus	The Occupancy status signal. (MV)
FirstOccupancy	The first occupancy initialisation after an Unoccupied period. (BV)
SblndTranslation	Sunblinds default translation command. (AV)
SblndRotation	Sunblinds default rotation command. (AV)
SblndUpdate	Sunblinds update command. (AV)
ResetBlindOvr	Reset manual sun blinds override command. (BV)
ResetFanOvr	Reset manual fan override command. (BV)
ResetSpOvr	Reset manual setpoint override command. (BV)
ResetLightOvr	Reset manual lights override command. (BV)

Default BACnet objects

The *RoomOccupancy (RoOc)* block related BACnet objects.

Output Parameter	Description
RoomOccupancy (BV100)	Holds information on current room occupancy status. BACnet Binary Value 100 input/output signal (can be modified inside of the block through the variable reference number)
OccupancyCmd (MV100)	Occupancy Command signal, usually coming from the schedule block. BACnet Multistate Value 100 input signal.
HoldTime (AV100)	Motion detection Holdup Time. BACnet Analogue Value 100 input signal.
UnoccOccDelay (AV101)	Transition delay from the Unoccupied to Occupied Status, forced by the Room Occupancy detection. BACnet Analogue Value 101 input signal.
BypassTime (AV102)	Bypass mode duration time. BACnet Analogue Value 102 input signal.
OccupancyStatus (MV101)	The Occupancy status signal. BACnet Multistate Value 100 output signal.

Block Functions

OccupancyStatus algorithm

The block analyses input signals to determine the unit's OccupancyStatus, which is used in different sections of the code to invoke corresponding actions (for example to switch effective temperature set-points occupied, standby and unoccupied modes). It is set by the OccupancyCmd and RoomOccupancy signals.

By default, the OccupancyStatus follows this algorithm matrix.

OccupancyCmd	+ RoomOccupancy ->	OccupancyStatus
Unoccupied	Unoccupied	Unoccupied

Unoccupied	Occupied	Bypass/Occupied
Bypass	Unoccupied	Unoccupied
Bypass	Occupied	Bypass/Occupied
Standby	Unoccupied	Standby
Standby	Occupied	Occupied
Occupied	Unoccupied	Standby
Occupied	Occupied	Occupied

RoomOccupancy

Room Occupancy is a binary status signal. It is used to represent the actual presence of occupants in the controlled space. It is activated when:

- OccDetection shows presence in the room;
- ManOccCmd is set to Occupied or Bypass state;
- RoomSensUpdt was sent due to a user action (for example a setpoint change);
- OccupancyCmd changes to Occupied and OccDetection is not available;
- WindowOpened changes to closed and the OccDetection is not available.

It is deactivated when:

- OccDetection shows no presence in the room for longer than the C.HoldTime parameter;
- ManOccCmd is set to Unoccupied or Standby state;
- The FCU is in a Bypass OccupancyState and a bypass delay elapses;
- OccupancyCmd changes to Unoccupied state, if the OccDetection is not available and the C.EnableRoomUnocc parameter is true;
- WindowOpened is active.

NOTE: The block interacts with the RoomOccupancy variable (by default BV100) via its number reference, which means that even though the variable itself is outside of the block it might be changed by the code inside it.

OccupancyCmd

The Occupancy Command signal is used to convey information from a Scheduler block. Normally it is set either to Occupied or Unoccupied. When it is set to:

- Unoccupied state – setting the Room Occupancy to Occupied may upgrade the effective Occupancy Status to Bypass/Occupied.
- Occupied state – setting the Room Occupancy to Unoccupied downgrades the effective Occupancy Status to Standby.

The Room Occupancy and Occupancy Status in the function the OccupancyCmd changes.

OccupancyCmd	OccDetection ->	RoomOccupancy	OccupancyStatus
Standby	Unconfig	-> set to Unoccupied	-> set to Standby
Standby	On	-> set to Occupied	-> set to Occupied
Standby	Off	-> set to Unoccupied after C.HoldTime	-> set to Standby after C.HoldTime
Bypass	Unconfig	-> set to Occupied. When C.BypassTime elapses, set to Unoccupied.	-> set to Bypass. When C.BypassTime elapses, set to Unoccupied.
Bypass	On	-> set to Occupied	-> set to Occupied after C.UnoccOccDelay
Bypass	Off	-> set to Unoccupied after C.HoldTime	-> set to Unoccupied after C.HoldTime

NOTE: By default, the OccupancyCmd and the ManOccCmd are not set to Standby nor Bypass states. These two states are usually a result of specific combinations of the OccupancyCmd, ManOccCmd and OccDetection assigned to the OccupancyStatus.

OccDetection

The OccDetection is used to connect a room motion sensor. It is a multistate value and can take the following states:

- On – presence is detected in the room.
- Off – no presence in the room.
- Unconfig – the motion sensor is disabled.

If a motion sensor is present, it sets the RoomOccupancy when the motion is detected and resets it when there is no motion in the controlled space.

Motion detection is maintained for a time defined by the C.HoldTime parameter when the OccupacyCmd is in the Occupied, Standby or Bypass mode, and for 1/3 of this time when the OccupacyCmd is Unoccupied.

If the motion sensor is present, the effective OccupancyStatus will not be set to the Bypass mode, it will be set to the Occupied mode instead.

If the motion sensor is not present, the OccupancyStatus will be set directly by changes of the OccCommand or ManOccCmd signals.

ManOccCmd

This signal is used to interface the OccupancyStatus with commands coming from users via room sensors, remote controls, mobile applications, etc. Its actions are event activated by the ManOccCmd.Updt signal.

If the OccupancyStatus is changed by other points, the ManOccCmd will synchronize with it after a foxed delay of 3s.

The Room Occupancy and the Occupancy Status in the function the ManOccCmd changes.

ManOccCmd	OccupancyCmd	OccDetection ->	RoomOccupancy	OccupancyStatus
Occupied	Unoccupied	Unconfig	-> set to Occupied When C.BypassTime elapses, set to Unoccupied.	-> set to Bypass When C.BypassTime elapses, set to Unoccupied.
Unoccupied	Unoccupied	Unconfig	-> set to Unoccupied	-> set to Unoccupied
Occupied	Unoccupied	On	-> set to Occupied	-> set to Unoccupied When the C.Unocc OccDelay elapses, set to Occupied.
Unoccupied	Unoccupied	On	-> set to Unoccupied When the C.Motion LockDelay elapses, set to Occupied.	-> set to Unoccupied When the C.Unocc OccDelay elapses, set to Occupied.
Occupied	Unoccupied	Off	-> set to Occupied	-> set to Occupied When the C.Hold Time elapses, set to Unoccupied.
Unoccupied	Unoccupied	Off	-> set to Unoccupied	-> set to Unoccupied
Occupied	Occupied	Unconfig	-> set to Occupied	-> set to Occupied.
Unoccupied	Occupied	Unconfig	-> set to Unoccupied	-> set to Standby
Occupied	Occupied	On	-> set to Occupied	-> set to Occupied
Unoccupied	Occupied	On	-> set to Unoccupied When the C.Motion LockDelay elapses, set to Occupied.	-> set to Standby When the C.Unocc OccDelay elapses, set to Occupied.

Occupied	Occupied	Off	-> set to Occupied	-> set to Occupied When the C.Hold Time elapses, set to Standby.
Unoccupied	Occupied	Off	-> set to Unoccupied	-> set to Standby

RoomSensUdt

The RoomSensUdt signal is activated when user interaction with a room sensor/control panel is detected. It sets the RoomOccupancy to occupied, and then the situation develops identically as per ManOccCmd changes.

WindowContact

The WindowContact signal is a multistate value. It can have three valid states:

1. Close – the algorithm follows its ordinary course.
2. Open – OccupancyStatus and the RoomOccupancy are set to Unoccupied.
3. Unconfig – WindowContact is disabled.

Fan speed override reset

When the RoomOccupancy goes into the Unoccupied state, or there is a change of the OccupancyStatus, the ResetFanOvr signal is sent to reset room control panel generated overrides of the fan speed.

Temperature setpoint override reset

When OccupancyCmd and RoomOccupancy are Unoccupied or there is a change of the OccupancyStatus, the ResetSpOvr signal is sent to reset room control panel generated overrides of the room temperature setpoint.

Lights control override reset

When the RoomOccupancy changes to the Occupied state or there is a change of the OccupancyStatus the ResetLightOvr signal is sent to reset room control panel generated overrides of the lights.

Sunblinds default values and override reset

When OccupancyCmd and RoomOccupancy are Unoccupied, the block will assign the C.UnoccTranslation and C.UnoccRotation to corresponding SblndTranslation and SblndRotation outputs and send the ResetBlindOvr signal to reset room control panel generated overrides of the blinds and bring them to default position.

When the OccupancyStatus changes, default occupancy period values represented by the C.OccTranslation and C.OccRotation are sent to the corresponding SblndTranslation and SblndRotation outputs.

The block also generates the SblndUpdate signal when both the OccupancyCmd and the RoomOccupancy go into the Unoccupied state, or they both go into the Occupied state for the first time after an Unoccupied period.

RoomSetpointValidation control block

Room Setpoint Validation control block (RoSpVa)

The *RoomSetpointValidation (RoSpVa)* block manages the room temperature setpoints in the function of the OccupancyStatus and HVACModeStatus of the unit.

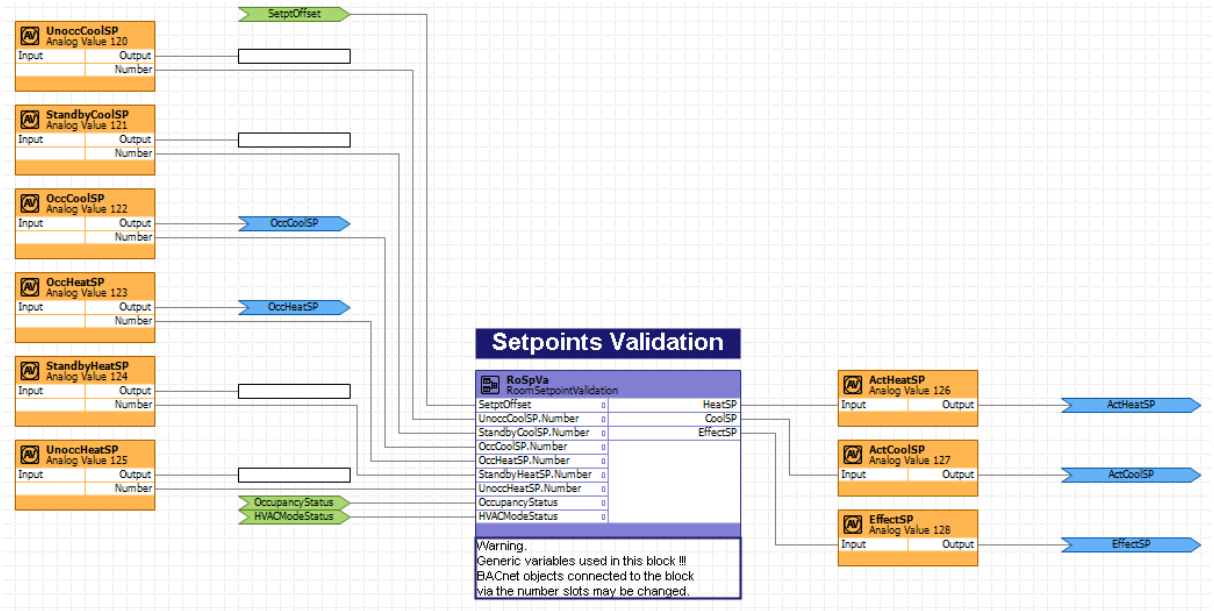


Figure 21: The RoomSetpointValidation (RoSpVa) programming block with corresponding variables.

Inputs

The *RoomSetpointValidation (RoSpVa)* block requires a connection to the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
SetptOffset	The Occupancy Status signal. (MV)
UnoccCoolSP.Number	Unoccupied cooling space temperature setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
StandbyCoolSP.Number	Standby cooling space temperature setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
OccCoolSP.Number	Occupied cooling space temperature setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
OccHeatSP.Number	Occupied heating space temperature setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
StandbyHeatSP.Number	Standby heating space temperature setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
UnoccHeatSP.Number	Unoccupied heating space temperature setpoint. (AV reference) WARNING. The block uses this point to modify the connected variable.
OccupancyStatus	The Occupancy Status signal. (MV)
HVACModeStatus	An HVAC mode status carrying information on the current mode of the unit. (MV)

Optional Inputs

The RoomSetpointValidation (RoSpVa) block does not use optional inputs

Configuration Parameters

Input Parameter	Description
C.UnoccStbyGap	Gap between Unoccupied and Standby setpoints. Default 0.2ΔK, 0.5ΔºF
C.OccGap	Gap between Occupied setpoints. Default 0.1ΔK, 0.2ΔºF
C.MinOccCool	Minimum Occupied Cooling setpoint. Default 15.5ºC, 60ºF
C.MaxOccCool	Maximum Occupied Cooling setpoint. Default 32.5ºC, 90ºF
C.MinOccHeat	Minimum Occupied Heating setpoint. Default 12.5ºC, 55ºF
C.MaxOccHeat	Maximum Occupied Heating setpoint. Default 29.5ºC, 85ºF
C.DefOccCool	Default Occupied Cooling setpoint. Default 23.5ºC, 75ºF
C.DefOccHeat	Default Occupied Heating setpoint. Default 21.5ºC, 71ΔºF
C.MaxSpOffset	Maximum value of the setpoint offset. Default 10ΔK, 18ΔºF
C.MinSpOffset	Minimum value of the setpoint offset. Default -10ΔK, -18ΔºF

Outputs

The *RoomSetpointValidation (RoSpVa)* block output signals (hardware and logic).

Output Parameter	Description
HeatSP	Active heating setpoint. (AV)
CoolSP	Active cooling setpoint. (AV)
EffectSP	An effective setpoint. (AV)

Default BACnet objects

The *RoomSetpointValidation (RoSpVa)* block related BACnet objects.

Output Parameter	Description
UnoccCoolSP (AV120)	Unoccupied cooling space temperature setpoint. BACnet Analogue Value 120 input/output signal (can be modified inside of the block through the variable reference number). Default value: 28ºC, 80ºF.
StandbyCoolSP (AV121)	Standby cooling space temperature setpoint. (AV reference) BACnet Analogue Value 121 input/output signal (can be modified inside of the block through the variable reference number). Default value: 26ºC, 78ºF.
OccCoolSP (AV122)	Occupied cooling space temperature setpoint. (AV reference) BACnet Analogue Value 122 input/output signal (can be modified inside of the block through the variable reference number). Default value: 23ºC, 75ºF.
OccHeatSP (AV123)	Occupied heating space temperature setpoint. (AV reference) BACnet Analogue Value 123 input/output signal (can be modified inside of the block through the variable reference number). Default value: 21ºC, 71ºF.
StandbyHeatSP (AV124)	Standby heating space temperature setpoint. (AV reference) BACnet Analogue Value 124 input/output signal (can be modified inside of the block through the variable reference number). Default value: 19ºC, 68ºF.
UnoccHeatSP (AV125)	Unoccupied heating space temperature setpoint. (AV reference) BACnet Analogue Value 125 input/output signal (can be modified inside of

Output Parameter	Description
	the block through the variable reference number). Default value: 16ºC, 66ºF.
ActHeatSP (AV126)	Active heating setpoint. BACnet Analogue Value 126 output signal.
ActCoolSP (AV127)	Active cooling setpoint. BACnet Analogue Value 127 output signal.
EffectSP (AV128)	An effective setpoint. BACnet Analogue Value 128 output signal.

Block Functions

Setpoint validation algorithm

The block uses the following algorithm to validate setpoint sequence:

UnoccCoolSP > StandbyCoolSP > OccCoolSP > OccHeatSP > StandbyHeatSP > UnoccHeatSP. The diagrams show the sequence for cooling setpoints. The heating setpoints follow the same logic with the only difference being that UnoccHeatSP < StandbyHeatSP < OccHeatSP.

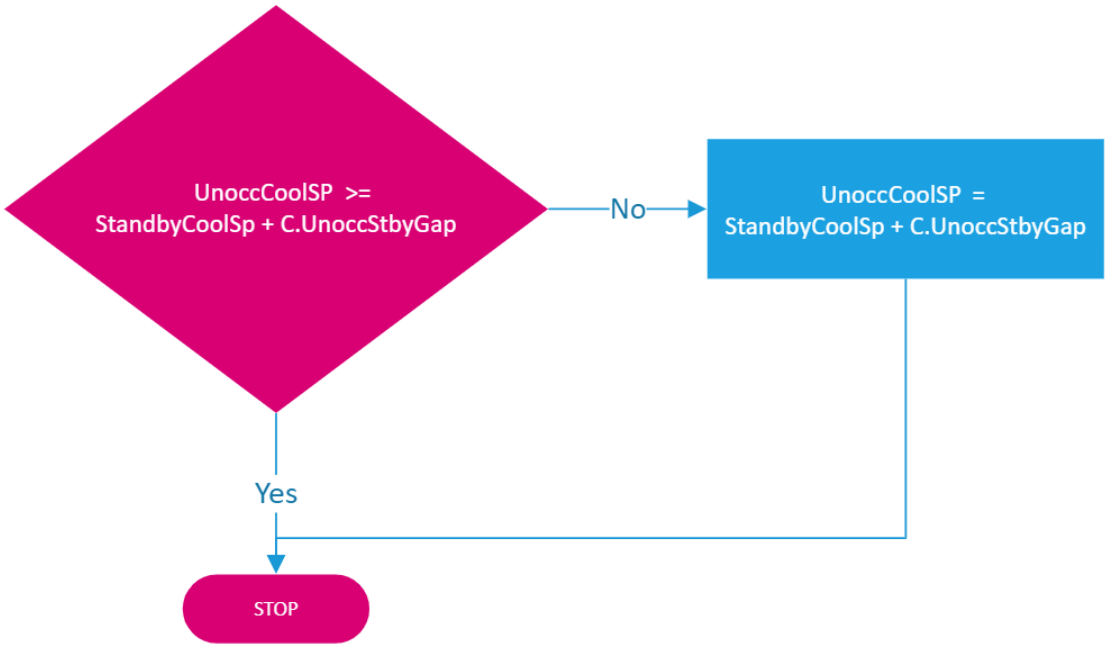


Figure 22: The UnoccCoolSP validation.

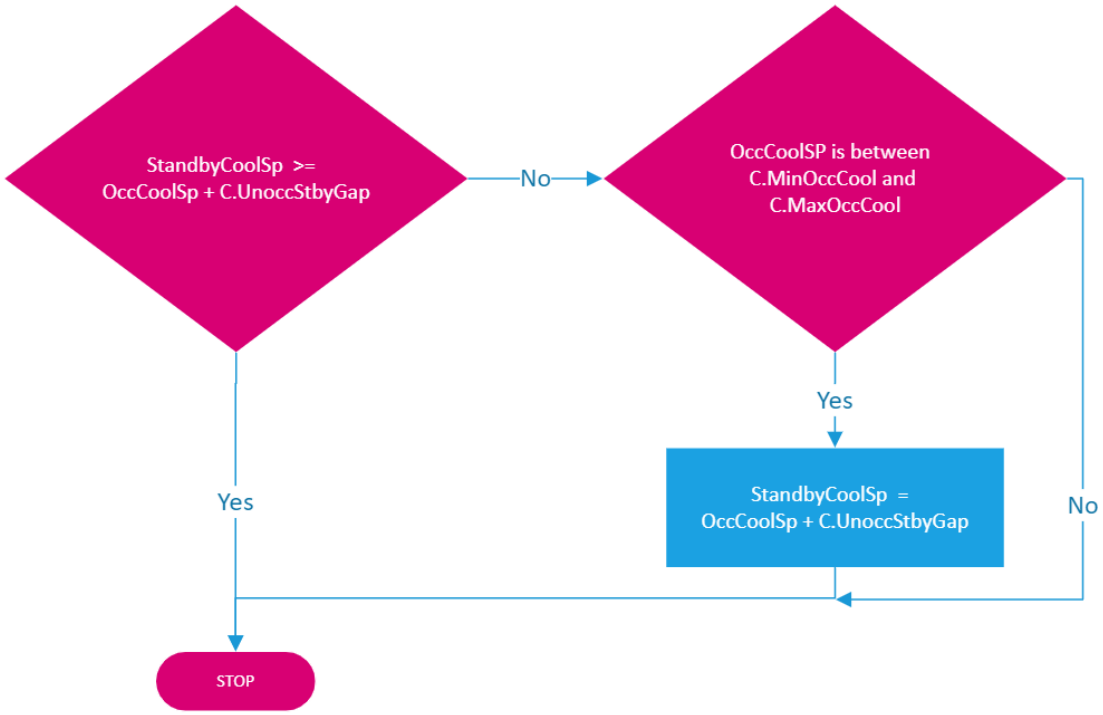


Figure 23: The StandbyCoolSP validation.

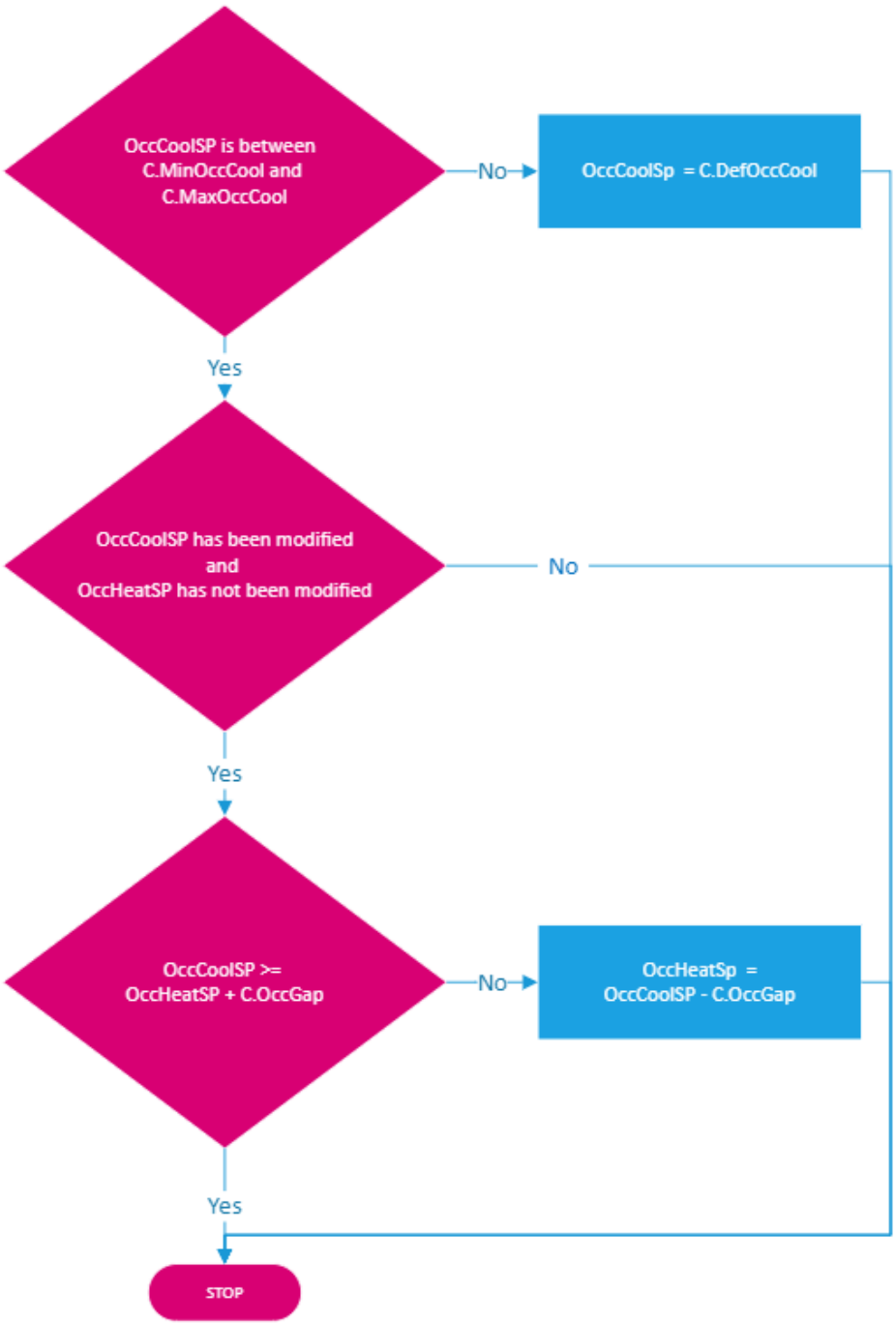


Figure 24: The OccCoolSP validation

If the block finds it necessary to modify the setpoint variables, it will use the reference number IDs to access and modify original BACnet objects.

Setpoint Offset algorithm

The SetptOffset input is used to shift heating and cooling setpoints when the OccupancyStatus input is in occupied, bypass or standby state. The SetptOffset usually comes from the room sensor/control panel. The SetptOffset is ignored if the OccupancyStatus is unoccupied or the HVACModeStatus input is in the MrngWrmup(3), NightPurge(6) or PreeCool(6) state.

OccupancyStatus ->	ActHeatSP	ActCoolSP
Unoccupied	UnoccHeatSP	UnoccCoolSP
Standby	StandbyHeatSP + SetptOffset	StandbyHeatSP + SetptOffset
Bypass	OccHeatSP + SetptOffset	OccHeatSP + SetptOffset
Occupied	OccHeatSP + SetptOffset	OccHeatSP + SetptOffset

Correlation between the OccupancyStatus and the ActHeat/CoolSP outputs.

Effective setpoint

When the HVACModeStatus is in the Heat(2) or MrngWrmup(3), the EffectSP output will be set to the ActHeatSP. For all the other HVACModeStatus values, the EffectSP will follow the ActCoolSP.

SeriesFanControl block

SeriesFanCtrl control block (SeFaCt)

The *SeriesFanCtrl (SeFaCt)* block operates a Series VAV fan.

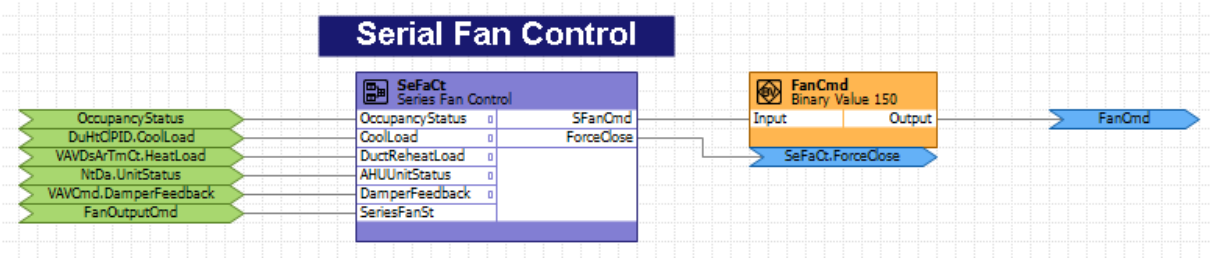


Figure 25: The SeriesFanCtrl (SeFaCt) programming block with corresponding variables.

Inputs

The *SeriesFanCtrl (SeFaCt)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
OccupancyStatus	Occupancy Status signal. (MV)
CoolLoad	Cooling load control signal. Usually, the signal comes from a DualHeatCoolPID control block. (AV)
DuctReheatLoad	Reheater load control signal. Usually, the signal comes from a VAVDischargeAirTemp control block. (AV)
UnitStatus	A superordinate AHU operational status signal. (BV)
DamperFeedback	A feedback signal from the damper actuator hardware output block. (AV)

Optional Inputs

Input Parameter	Description
SeriesFanSt	Fan operation feedback signal. The signal usually comes from the hardware output feedback output. Default *null* (BV)

Outputs

The *SeriesFanCtrl (SeFaCt)* block output signals (hardware and logic).

Output Parameter	Description
FanCmd	Fan start signal. (BV)
ForceClose	Damper closed override command. Should be connected to a ForceClose input of a DamperControl block. (BV)

Default BACnet objects

The *SeriesFanCtrl (SeFaCt)* block related BACnet objects.

Output Parameter	Description
FanCmd (BV 150)	Fan start command signal. BACnet Binary Value 150 output signal.

Block Functions

Series fan start algorithm

The series fan is started if any of the following occur:

- the VAV is in the Occupied or Bypass OccupancyStatus;
- the DuctReheatLoad is higher than 10% (a 10% switch-off hysteresis applies);
- the CoolLoad is higher than 50% (a 50% switch-off hysteresis applies) and UnitStatus is active;

When the start conditions are fulfilled, the block sends ForceClose signal, which needs to be connected to a ForceClose input of a DamperPosition control block, to force closing of the VAV damper. It is done to prevent reversed rotation of the fan by the airflow coming from the VAV. Once the SeriesFanCtrl block sees that DamperFeedback input is below 1% it activates FanCmd output to start the fan. Minimum on(120s) and off(120s) time block prevents the fan to be switched on/off too frequently.

ValveExercise block

Valve Exercise control block (VlEi)

The *ValveExercise (VlEi)* block calculates heating and cooling load signals required to keep the space temperature at corresponding heating and cooling setpoints.

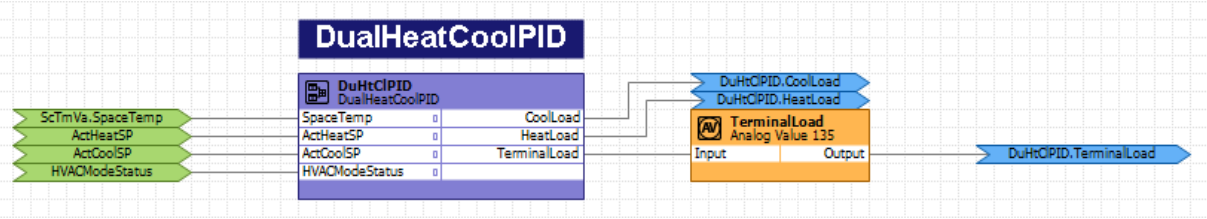


Figure 26: The ValveExercise (VlEi) programming block with corresponding variables.

Inputs

The *ValveExercise (VlEi)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
ValveCtrl	Valve control signal. (AV)
OccupancyStatus	The occupancy status of the controlled space. (MV)

Configuration Parameters

Input Parameter	Description
C.ValveIdleDelay	Valve idle delay before the exercise function can be started. Default *15 days*(AV)
C.UnoccDelay	Minimum unoccupied mode time before the exercise function can be started. Default *6h* (AV)
C.ExercTime	The exercise function duration. Default *300s* (AV)
C.ValveExercPos	The valve exercise position. Default *100%* (AV)

Outputs

The *ValveExercise (VlEi)* block output signals (hardware and logic).

Output Parameter	Description
Valve	An effective valve control signal. (AV) (0%-100%)

Block Functions

Valve exercise

The valve exercise function is activated when the ValveCtrl is inactive (equal to 0%) for at least C.ValveIdleDelay and the Occupancy is Unoccupied(2) for at least C.UnoccDelay.

Once started the valve exercise will force the Valve output to the C.ValveExecPos value for a time defined by the C.ExercTime parameter. If at any moment during the valve exercise action the ValveCtrl input activates, the exercise action is ended and normal control takes over the Valve output.

VAVDischargeAirTemp control block

VAV Discharge Air Temperature control block (VAVDsArTmCt)

The *VAVDischargeAirTempControl (VAVDsArTmCt)* block calculates a duct heater heating load and VAV heating load in the function of the DischaAirTemp.

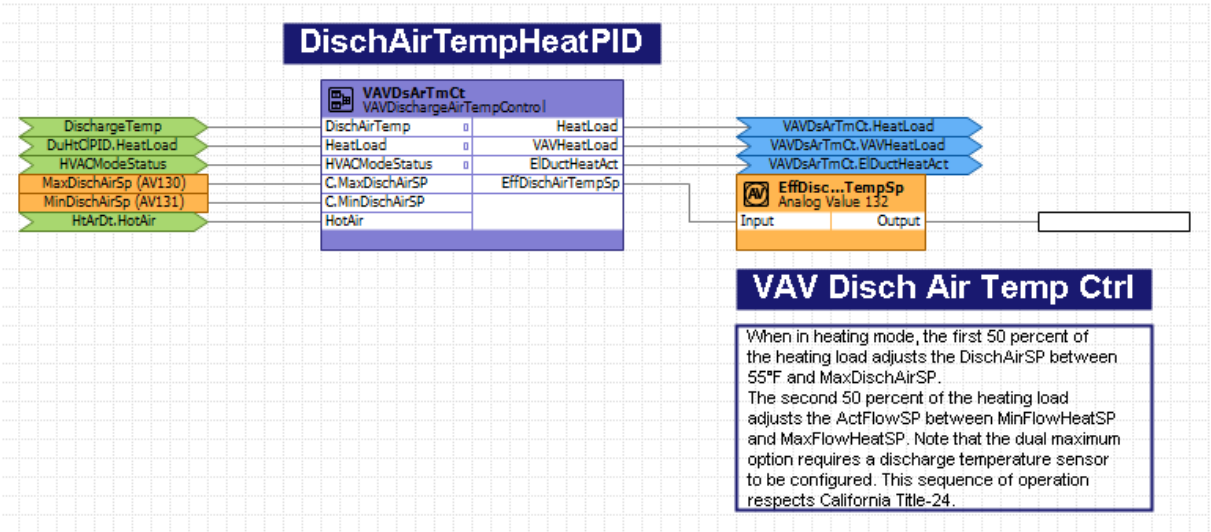


Figure 27: The VAVDischargeAirTempControl (VAVDsArTmCt) programming block with corresponding variables.

Inputs

The *VAVDischargeAirTempControl (VAVDsArTmCt)* block takes the following input signals (hardware and logic).

Required Inputs

Input Parameter	Description
DischAirTemp	Effective measured Discharge Air Temperature value. (AV)
HeatLoad	Active heating demand signal. (AV)
HVACModeStatus	An HVAC mode status input carrying information on the current mode of the unit. (MV)

Optional Inputs

Input Parameter	Description
HotAir	A confirmation signal, that the air temperature supplied from an AHU to the VAV's intake is high enough to use it for heating. Default *true* (BV)

Configuration Parameters

Input Parameter	Description
C.MinDischAirSP	Minimum value of the discharge air temperature setpoint. Default: 12.8ºC, 55ºF (AV)
C.MaxDischAirSP	Maximum value of the discharge air temperature setpoint. Default: 33ºC, 95ºF (AV)

Outputs

The *VAVDischargeAirTempControl (VAVDsArTmCt)* block output signals (hardware and logic).

Output Parameter	Description
HeatLoad	An effective heating demand signal. (AV) (0%-100%)
VAVHeatLoad	A heating demand signal used to control the VAV's flow. (AV) (0%-100%)
ElDuctHeatAct	A signal used to force a minimum airflow in case an electrical heater is used. (BV)
EffDischAirTempSp	An effective discharge air temperature setpoint. (AV ºC/ºF)

Default BACnet objects

The *VAVDischargeAirTempControl (VAVDsArTmCt)* block related BACnet objects.

Output Parameter	Description
MinDischAirSP (AV130)	A minimum value of the discharge air temperature setpoint. BACnet Analogue Value (AV130) input signal.
MaxDischAirSP (AV131)	A maximum value of the discharge air temperature setpoint. BACnet Analogue Value (AV131) input signal.
EffDischAirTempSp (AV132)	An effective discharge air temperature setpoint. BACnet Analogue Value (AV132) output signal.

Block Functions

Operation principle

When in heating mode, the lower 50% of the heating load input signal modulates the DischAirSP between the MinDischAirSP and MaxDischAirSP. The calculated value is then exposed via the EffDischAirTempSp and used internally by a discharge air temperature control algorithm.

The higher 50% of the heating load sets the ActFlowSP between the MinFlowHeatSP and the MaxFlowHeatSP. This value is then used to drive the VAV's airflow in heating mode.

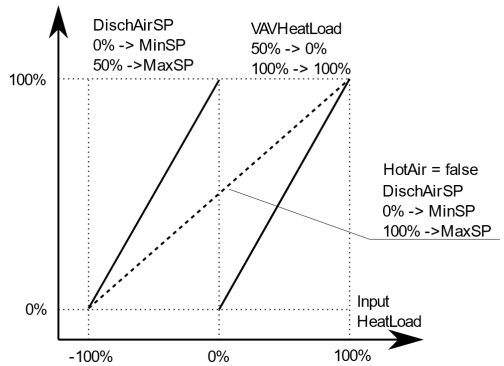


Figure 28: The VAVDischargeAirTempControl (VAVDsArTmCt) operation principle.

NOTE: The dual maximum option requires a discharge temperature sensor to be available. This sequence of operation respects California Title-24 regulation.

Hot Air detection

The HotAir input signal is activated when the air temperature supplied from an AHU at the VAV's intake is sufficiently high for heating. If the HotAir is false, the VAV flow control block will ignore the VAVHeatLoad, and thus the whole range of the HeatLoad input signal will be used to calculate the DischAirSP using the MinDischAirSP and the MaxDischAirSP (0% to 100% of the HeatLoad, instead of from 0% to 50%).

Heating PID activation

The block uses a BACnet PID Loop object to control heating action. It takes in the DischAirTemp as an input value and the EffDischAirTempSp as a setpoint value. It is activated when the HVACModeStatus is in one of the following states Auto(1), Heat(2), MrngWrmUp(3), Test(8), EmergHeat(9), MaxHeat(13), Economy (14) or Dehumid(15). Its output (the HeatLoad output signal) is used to control duct heaters.

Electrical heater activation output

If an electrical duct heater is used, the VAV flow control block must be set to minimum flow value to prevent overheating in case the heater is activated, and no flow is present. The ElDuctHeatAct output is set when the HeatLoad output signal exceeds 10% and reset when it drops to 0%. When connected to a VAV control block, activation of the ElDuctHeatAct forces the airflow to be no less than a minimum heating mode airflow.