# FCU Systems

User Guide

DISTECH CONTROLS™

# Table of Contents

# 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.

# 3SpeedFanControl block

## 3SpeedFanCtrl control block (3SdFnCt)

The *3SpeedFanCtrl (3SdFnCt)* block operates a three-speed fan.

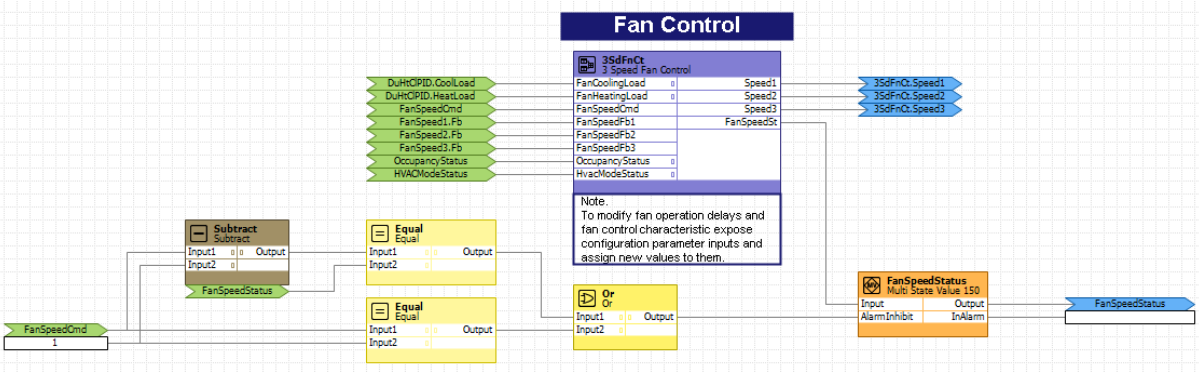


*Figure 1: The 3SpeedFanCtrl (3SdFnCt) programming block with corresponding variables.*

### Inputs

The *3SpeedFanCtrl (3SdFnCt)* block takes the following input signals (hardware and logic).

#### Required Inputs

| Input Parameter | Description |
|---|---|
| FanCoolLoad | A cooling demand signal. (AV) |
| FanHeatLoad | A heating demand signal. (AV) |
| FanSpeedFb1 | Fan speed 1 output feedback signal. (BV) |
| FanSpeedFb2 | Fan speed 2 output feedback signal. (BV) |
| FanSpeedFb3 | Fan speed 3 output feedback signal. (BV) |
| OccupancyStatus | Occupancy Status signal. (MV) |
| HVACModeStatus | An effective HVAC mode status of the unit. (MV) |

#### Optional Inputs

| Input Parameter | Description |
|---|---|
| FanSpeedCmd | Fan speed command (status) usually from a room sensor/control panel. Default *Auto* (MV) |
| EnblRetAirSample | Enable return air sampling. Default *false* (BV) |

#### Configuration Parameters

| Input Parameter | Description |
|---|---|
| C.FanCoolCtrlMode | Fan cooling control mode (AlwaysOn(1), OnDemand(2), Smart(3), Off(4)). Default *Smart(3)*(MV) |
| C.FanHeatCtrlMode | Fan Heating control mode (AlwaysOn(1), OnDemand(2), Smart(3), Off(4)). Default *Smart(3)*(MV) |
| C.FanOffDelay | Fan stop delay. Default *180s*(AV) |
| C.FanDfltSpeed | Fan default speed (active for FanOnly(10), FreeCool(12) and Ice(13) HVACModeStatuses). Default *100%*(AV) |
| C.FanSpeed1Level | Fan speed 1 threshold. Default: *5%*(AV) |

| Input Parameter | Description |
|---|---|
| C.FanSpeed2Level | Fan speed 2 threshold. Default: 33 %(AV) |
| C.FanSpeed3Level | Fan speed 3 threshold. Default: 67 %(AV) |

## Outputs

The *3SpeedFanCtrl (3SdFnCt)* block output signals (hardware and logic).

| Output Parameter | Description |
|---|---|
| Speed1 | Fan speed 1 start signal. (BV) |
| Speed2 | Fan speed 2 start signal. (BV) |
| Speed3 | Fan speed 3 start signal. (BV) |
| FanSpeedStatus | A multistate interface point (Off(1). Low(2), Med(3), High(4)). (MV) |
| EnableEquip | Signal used to enable equipment requiring fan operation. (BV) |

## Default BACnet objects

The *3SpeedFanCtrl (3SdFnCt)* block related BACnet objects.

| Output Parameter | Description |
|---|---|
| FanSpeedStatus (MV 150) | A multistate interface point.<br>BACnet Multistate Value 150 output signal. |

## Block Functions

### *Variable-speed fan control algorithm*

The variable speed fan can have four different operating modes defined separately for heating and cooling actions by the C.FanHeatCtrlMode and the C.FanCoolCtrlMode. They can be defined as:

1. AlwaysOn – the actual fan speed is determined as a higher value of the C.FanSpeed1Level and the value calculated from the FanHeat/CoolLoad;
2. OnDemand – the fan speed follows demand derived from the FanHeat/CoolLoad;
3. Smart – the fan speed follows the AlwaysOn scheme when the OccupancyStatus is Occupied(1) or Bypass(3) and on-demand when the OccupancyStatus is Unoccupied(2) or Standby(4);
4. Off – the fan speed is set to 0%.

### *HVACModeStatus impact*

The effective heating/cooling fan speed values are filtered by the HVACModeStatus. When the HVAC-ModeStatus is equal to:

- Auto(1), Test(8), Economy(14), Dehumid(15) – the fan speed is controlled by the higher of the heating and cooling signals;
- Heat(2), MrngWarmUp(3), MaxHeat(13) – the fan speed follows the heating load signal;
- Cool(4), NightPurge(5), PreCool(6), FreeCool(11), Ice(12) – the fan speed follows the cooling load signal;
- Off(7) – the fan speed is set to 0%;
- EmergHeat(9) – the fan speed is set to 100%;
- FanOnly(10) – the fan speed is set to C.FanDfltSpeed;

## *FanSpeedCmd input*

The FanSpeedCmd usually is connected to a room control sensor/panel and enables users to override the output fan speed. It is a multistate signal and can take the following values:

1. Auto – the FanSpeedCtrl output follows heating/cooling load signals;
2. Off – the fan is stopped;
3. Low – the fan runs at speed 1;
4. Med – the fan runs at speed 2;
5. High – the fan runs at speed 3.

## *Freeze protection and HCACMode Off status*

If the HVACModeStatus is set to Off(7) the fan is switched off, unless the freeze protection mode is activated.

If the room temperature drops considerably, the HVACMode control block activates the EmergHeat(9) HVACModeStatus (for more information, refer to the HVACMode FCU control block section). The *3SpeedFanCtrl* block then checks whether the C. FanHeatCtrlMode is not equal to AlwaysOff(4) and if it is not, the fan speed 3 is activated.

The freeze protection mode has the highest priority of all the overrides.

## *FanSpeedStatus output*

The FanSpeedStatus output transfers the actual FanSpeedCtrl output per cent value to a three-speed multistate value, used commonly on room control sensors/panels and on supervisory systems' graphic pages. The FanSpeedStatus uses the following switch on/off points:

- Low speed:
  - switch-on point: higher of the C.FanSpeed1Level and a fixed 5% value;
  - switch-off point: the switch-on point – 3%;
- Medium speed:
  - switch-on point: the C.FanSpeed2Level;
  - switch-off point: higher of (the C.FanSpeed2Level – 5%) and the Low-speed switch on point;
- High speed:
  - switch-on point: the C.FanSpeed3Level;
  - switch-off point: higher of (the C.FanSpeed3Level – 5%) and the Medium-speed switch on point;

When the internally calculated fan control signal indicates that the fan should be switched off (none of the stages is active), the FanSpeedStatus output will be upheld at low speed for the time defined by the C.FanOffDelay parameter. During this time the fan operates at low speed.

## *Fan Speed outputs*

Fan speed outputs follow the internal control signal represented by the FanSpeedStatus. The output will be switched on with a fixed 1s delay, the internal switch on command is activated, and the feedback signals from the other two outputs are deactivated. It prevents simultaneous closing of two relays and a potential short circuit.

## *EnableEquip output*

Certain components of the FCU systems must not be turned on when the fan is not running and should be switched off before the fan is stopped (for example an electrical heater or a direct expansion cooling coil). The EnableEquip output is activated 10s after the FanEnableCmd and is switched off directly when the internally calculated fan control signal indicates, that the fan should be switched off (none of the stages is active), even though the actual fan operation is upheld for the time defined by the C.FanOffDelay parameter. It gives time for an electrical heater or a DX cooler to dissipate cumulated energy.

## *EnblRetAirSample activation*

The EnblRetAirSample comes into action when an FCU uses a return temperature sensor rather than a room temperature sensor. This means that the FCU's fan must be switched on periodically to sample the room temperature even in unoccupied mode. When the algorithm decides that a sample is required it activates the EnblRetAirSample. Then the *3SpeedFanCtrl* block checks whether the effective FanSpeedCmd is in the Auto(1) mode and forces the fan's speed to no less than the C.FanEnableThresh to extract room air and detect its temperature via the return air temperature sensor.

# Cooling Control block

## CoolingControl block (ClCt)

The *CoolingControl (ClCt)* block controls operation of a cooling coil.

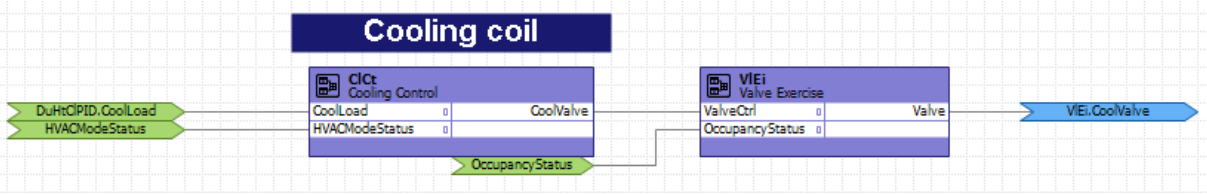


*Figure 2: The CoolingControl (ClCt) programming block with corresponding variables.*

### Inputs

The *CoolingControl (ClCt)* block takes in the following input signals (hardware and logic).

#### Required Inputs

| Input Parameter | Description |
|---|---|
| CoolLoad | A cooling demand signal. (AV) |
| HVACModeStatus | An HVAC mode status input carrying information on the current mode of the unit. (MV) |

### Outputs

The *CoolingControl (ClCt)* block output signals (hardware and logic).

| Output Parameter | Description |
|---|---|
| CoolValve | A cooling valve control signal. (AV) (0%-100%) |

### Block Functions

#### *Cooling valve activation*

The CoolValve signal is enabled when the HVACModeStatus is in Auto(1), Cool(4), NightPurge(5), PreCool(6), Test(8), Ice(12), Economy(14) or Dehumid(15) states. When disabled, the output is set to 0%, when enabled the signal follows the CoolLoad input with application of the ratio block characteristic.

#### *Ratio block calculation*

The ratio block allows driving multiple coolers from the same CoolLoad signal. By default, the block uses fixed constant values, but these can be easily modified. The abscissa values X1 and X2 enable assignment of any range of the CoolLoad to the full extent of the CoolValve output. The ordinate values Y1 and Y2 allow setting minimum and maximum value of the output signal.

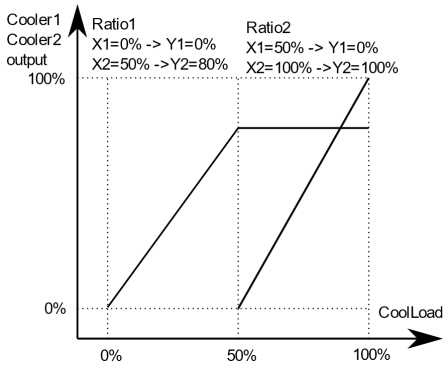


*Figure 3: An example of using the ratio blocks to drive two cooling coils with one CoolLoad signal*

# DischargeAirTemp control block

## Discharge Air Temperature control block (DsArTmCt)

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

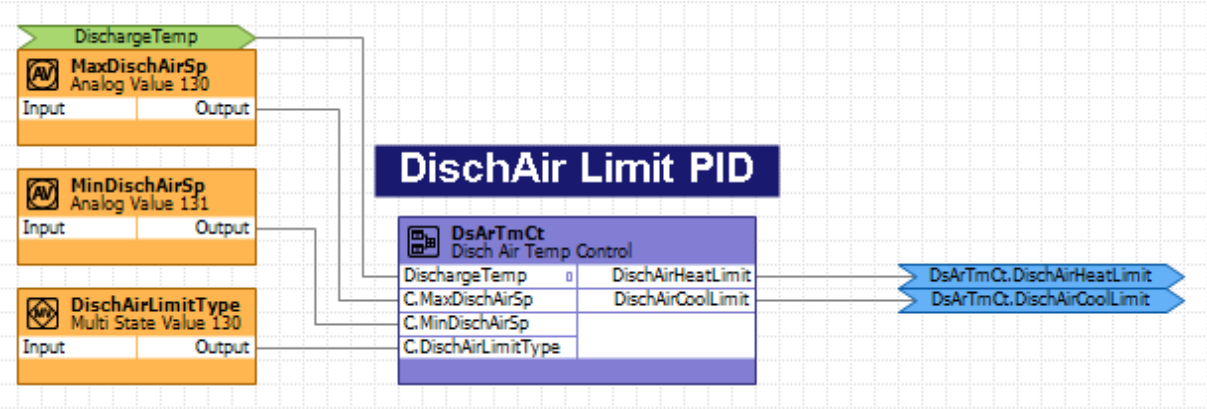


*Figure 4: The DischargeAirTempControl (DsArTmCt) programming block with corresponding variables.*

### Inputs

The *DischargeAirTempControl (DsArTmCt)* block takes in the following input signals (hardware and logic).

#### Required Inputs

| Input Parameter | Description |
|---|---|
| DischAirTemp | Effective measured Discharge Air Temperature value. (AV) |

#### Configuration Parameters

| Input Parameter | Description |
|---|---|
| C.MinDischAirSP | Minimum value of the discharge air temperature setpoint. Default: *16ºC, 73ºF* (AV) |
| C.MaxDischAirSP | Maximum value of the discharge air temperature setpoint. Default: *32ºC, 90ºF* (AV) |
| C.DischAirLimitType | Discharge air temperature limit type. Default: 32ºC, 95ºF (AV) |

### Outputs

The *DischargeAirTempControl (DsArTmCt)* block output signals (hardware and logic).

| Output Parameter | Description |
|---|---|
| DischAirHeatLimit | The current value of a Discharge Air Temp based heating action limitation. Default *100%* (AV) |
| DischAirCoolLimit | The current value of a Discharge Air Temp based cooling action limitation. Default *100%* (AV) |

### Default BACnet objects

The *DischargeAirTempControl (DsArTmCt)* block related BACnet objects.

| Output Parameter | Description |
|---|---|
| MinDischAirSP (AV130) | A minimum value of the discharge air temperature setpoint.<br>BACnet Analogue Value (AV130) input signal. |
| MaxDischAirSP (AV131) | A maximum value of the discharge air temperature setpoint.<br>BACnet Analogue Value (AV131) input signal. |
| DischAirLimitType (MV130) | Discharge air limit functional model.<br>BACnet Analogue Value (MV130) output signal. |

# Block Functions

## *Operation principle*

When the DischargeTemp sensor is present (its value is not null), the block will use two PID controllers and the C.DischAirLimitType parameter to generate limit signals for heating and cooling action when the measured temperature exceeds the C.MinDischAirSP or C.MaxDischAirSP. These signals are used by the main temperature controller to limit Heat/CoolLoad signals.

## *Limit type*

The C.DischAirLimitType configuration variable decides which of the limits will be taken into account. It can take following values:

1. NotLimited – limits are inactive;
2. HighLimit – only the C.MaxDischAirSp will be used to generate the DischAirHeatLimit output signal;
3. LowLimit – only the C.MinDischAirSp will be used to generate the DischAirCoolLimit output signal;
4. HighLowLimit – both the C.MaxDischAirSp and the C.MinDischAirSp will be used to generate the DischAirHeat/CoolLimit output signals.

When the limit is not active the corresponding output will be set to 100% to enable the use of the full range of the Heat/CoolLoad signal by the main temperature controller (usually DualHeatCoolPID block is used for that purpose).

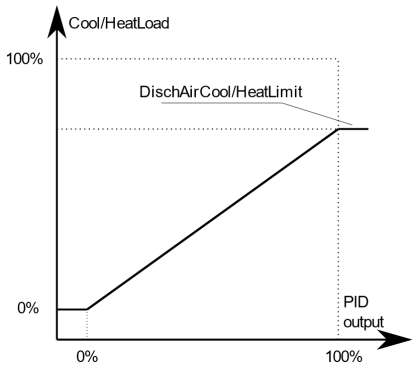


*Figure 5: DischAirCool/HeatLimit impact on the Cool/HeatLoad signal.*

# ECOVueFCU block

## ECO-Vue Calculation block for FCU (ECVeCl)

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

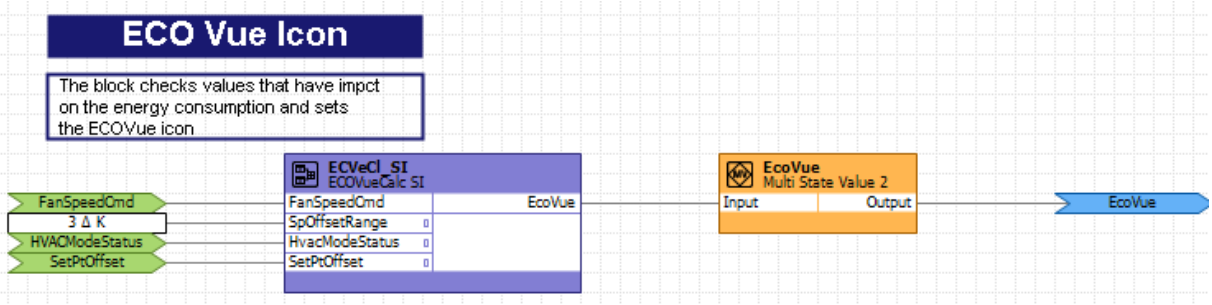


*Figure 6: The ECOVueCalculation (ECVeCl) programming block with corresponding variables.*

### 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 ECO-Vue. 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.

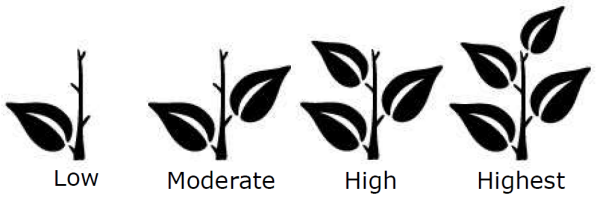


*Figure 7: EcoVue energy efficiency levels.*

## ECO-Vue calculation

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

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

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

# Heating Control block

## HeatingControl block (HtCt)

The *HeatingControl (HtCt)* block controls the operation of a heating coil.

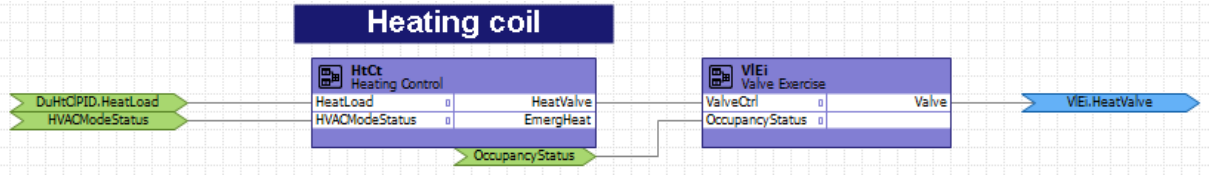


*Figure 8: The HeatingControl (HtCt) programming block with corresponding variables.*

### Inputs

The *HeatingControl (HtCt)* block takes in 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) |

### Outputs

The *HeatingControl (HtCt)* block output signals (hardware and logic).

| Output Parameter | Description |
|---|---|
| HeatValve | A heating valve control signal. (AV) (0%-100%) |

### Block Functions

#### *Heating valve activation*

The HeatValve signal is enabled when the HVACModeStatus is in Auto(1), Heat(2), MrngWrmUp(3), Test(8), EmergHeat(9), MaxHeat(13), Economy (14) or Dehumid(15) state. When disabled, the output is set to 0%, when enabled the signal follows the HeatLoad input with application of the ratio block characteristic.

#### *Emergency heat activation*

When the HVACModeStatus is in the EmergHeat(9) state, the HeatValve output is forced to 100%.

#### *Ratio block calculation*

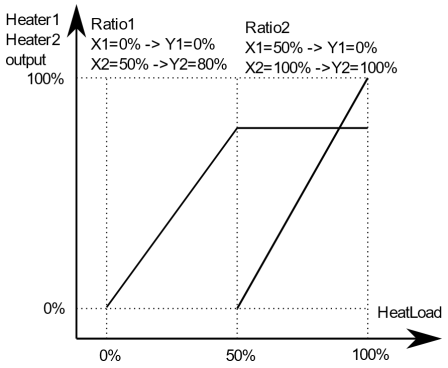


*Figure 9: An example of using the ratio blocks to drive two heaters with one HeatLoad signal.*

The ratio block allows driving multiple Heaters from the same HeatLoad signal. By default, the block uses fixed constant values, but these can be easily modified. The abscissa values X1 and X2 enable assignment of any range of the HeatLoad to the full extent of the HeatValve output. The ordinate values Y1 and Y2 allow setting minimum and maximum value of the output signal.

# HVAC Mode FCU control block

## HVAC Mode control block (HVACMd)

The *HVACMode (HVACMd)* block manages the operating mode of the unit.

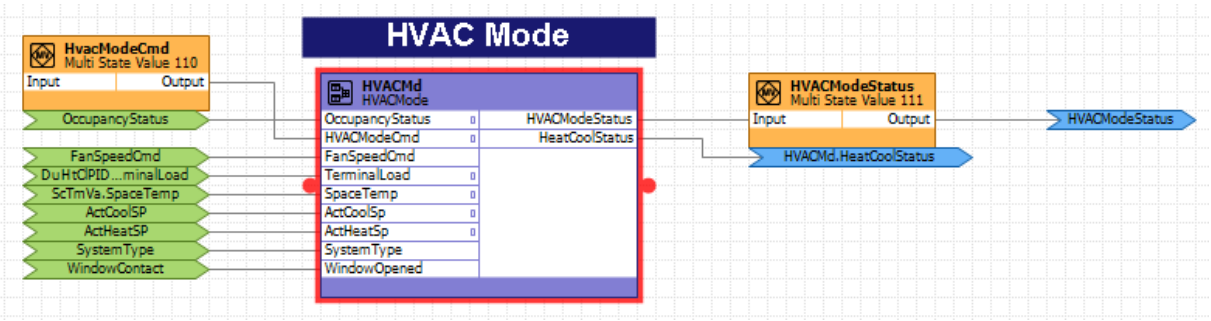


*Figure 10: The HVACMode (HVACMd) programming block with corresponding variables.*

### Inputs

The *HVACMode (HVACMd)* block requires connection of 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) |
| 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: *30s*(AV) |

## Outputs

The *HVACMode (HVACMd )* 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 *HVACMode (HVACMd )* block related BACnet objects.

| Output Parameter | Description |
|---|---|
| HVACModeCmd (MV110) | An HVAC mode command usually coming from a network or a supervisory system.<br>BACnet Multistate Value 110 output signal. |
| 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.<br>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:

1. AUTO Controller automatically determines an effective mode
2. HEAT Heating only
3. MRNG_WRMUP Application-specific morning warm-up
4. COOL Cooling only
5. NIGHT_PURGE Application-specific night purge
6. PRE_COOL Application-specific pre-cool
7. OFF Unit stopped, only safety actions active
8. TEST Equipment being tested
9. EMERG_HEAT Emergency heat mode (used for an anti-freeze action)
10. FAN_ONLY Air not conditioned; the fan turned on
11. FREE_COOL Cooling with cooling coil stopped
12. ICE Ice-making mode
13. MAX_HEAT Maximum heating mode
14. ECONOMY Economic Heat/Cool mode
15. 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 reinitialize control blocks in other sections of the application. Similar action is undertaken and upheld for 60s after controller restart.

## *Effective HVACModeCmd*

An effective internal HVACModeCmd follows the input value, except when:

- the WindowOpened signal indicates an opened window – the HVACModeCmd input 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, but 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 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, that means when one is enabled, the other is locked and cannot be switched on until the former is deactivated. To avoid going from cooling into heating and back in a rapid succession a changeover delay is introduced. When any of the abovementioned states deactivates, the other will go into action not 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 procedures. 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 UNITOUCH and EC-Multi-Sensor BLE devices. The output will be set to:

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

# Outside Air Damper control block

## Outside Air Damper block (OuArDm)

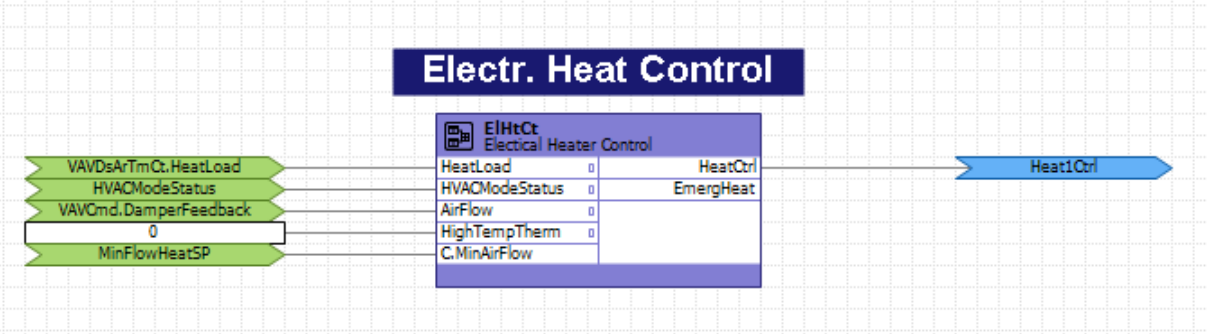


*Figure 11: The OutsideAirDamper (OuArDm) programming block with corresponding variables.*

The *OutsideAirDamper (OuArDm)* block controls an outside air damper's position in the function of CO2 and cooling load signals.

### Inputs

The *OutsideAirDamper (OuArDm)* block takes in the following input signals (hardware and logic).

#### Required Inputs

| Input Parameter | Description |
|---|---|
| CoolLoad | A cooling demand signal. (AV) |
| CO2Load | A CO2 concentration control signal. (AV) |
| OccupancyStatus | The Occupancy Status signal. (MV) |
| HVACModeStatus | An HVAC mode status input carrying information on the current mode of the unit. (MV) |

#### Configuration Parameters

| Input Parameter | Description |
|---|---|
| C.MaxPosition | Maximum damper position. Default *100%*(AV) |
| C.MinPosition | Minimum damper position. Default *10%*(AV) |

### Outputs

The *OutsideAirDamper (OuArDm)* block output signals (hardware and logic).

| Output Parameter | Description |
|---|---|
| DamperLoad | A damper control signal. (AV) (0%-100%) |

# Block Functions

## *Damper activation*

The DamperLoad is enabled when the HVACModeStatus is any state except for Off(7) and EmergHeat(9) and the OccupancyStatus is not Unoccupied. When disabled the damper is set to 0%.

## *Minimum damper position calculation*

When the OccupancyStatus is not Unoccupied, and the HVACModeStatus is set to FreeCool(11), the effective minimum damper position is based on the CoolLoad input and set by the following ratio characteristic:

When the OccupancyStatus is Unoccupied or the HVACModeStatus is different from Off(7), EmergHeat(9) or FreeCool(11), the effective minimum damper position is set to the C.MidPosition parameter.

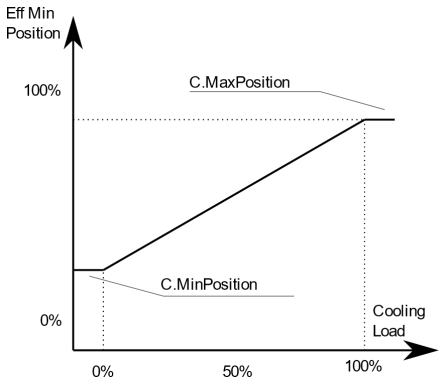


*Figure 12: Calculation of the effective minimum damper position.*

## *Damper control output*

When the OccupancyStatus is Occupied or Bypass and the HVACModeStatus is not set to Off(7) or EmergHeat(9), the DamperLoad output follows this ratio characteristic:

When the OccupancyStatus is Unoccupied or Standby the DamperLoad is set to the effective minimum damper position.

When HVACModeStatus is not set to Off(7) or EmergHeat(9) the DamperLoad is set to 0%.

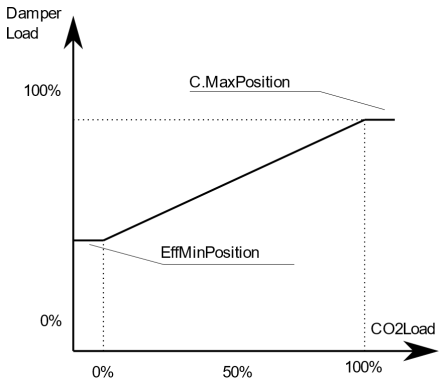


*Figure 13: Calculation of the DamperLoad output signal*

# VariableSpeedFanControl block

## VariableSpeedFanCtrl control block (VrSdFnCt)

The *VariableSpeedFanCtrl (VrSdFnCt)* block operates a variable speed fan.

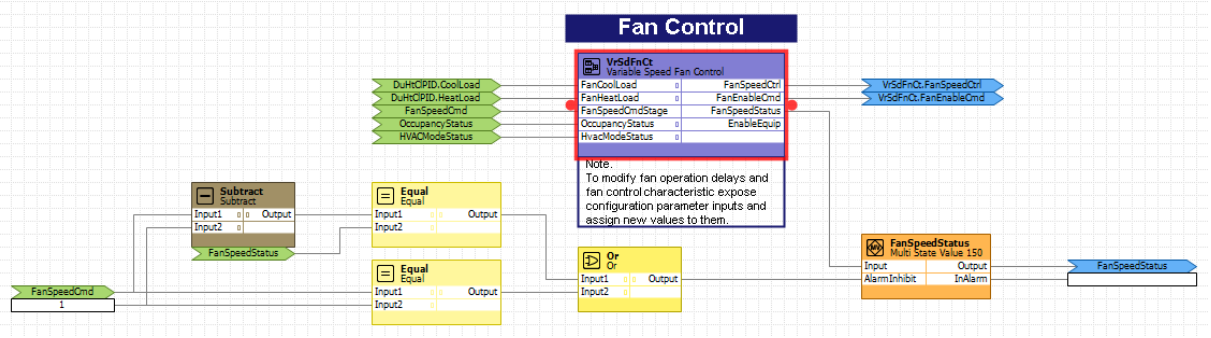


*Figure 14: The VariableSpeedFanCtrl (VrSdFnCt) programming block with corresponding variables.*

### Inputs

The *VariableSpeedFanCtrl (VrSdFnCt)* block takes in following input signals (hardware and logic).

#### Required Inputs

| Input Parameter | Description |
|---|---|
| FanCoolLoad | A cooling demand signal. (AV) |
| FanHeatLoad | A heating demand signal. (AV) |
| OccupancyStatus | Occupancy Status signal. (MV) |
| HVACModeStatus | An effective HVAC mode status of the unit. (MV) |

#### Optional Inputs

| Input Parameter | Description |
|---|---|
| FanSpeedCmd | Fan speed command (status) usually from a room sensor/control panel. Default *Auto* (MV) |
| FanSpeedCmdPct | Fan speed command (per cent) usually from a room sensor/control panel. Default *null* (MV) |
| EnblRetAirSample | Enable return air sampling. Default *false* (BV) |

#### Configuration Parameters

| Input Parameter | Description |
|---|---|
| C.FanCoolCtrlMode | Fan cooling control mode (AlwaysOn(1), OnDemand(2), Smart(3), Off(4)). Default *Smart(3)*(MV) |
| C.FanHeatCtrlMode | Fan Heating control mode (AlwaysOn(1), OnDemand(2), Smart(3), Off(4)). Default *Smart(3)*(MV) |
| C.FanEnableThresh | Fan enable threshold. Default: *10%*(AV) |
| C.FanOffDelay | Fan stop delay. Default *180s*(AV) |
| C.FanDfltSpeed | Fan default speed (active for FanOnly(10), FreeCool(12) and Ice(13) HVACModeStatuses). Default *100%*(AV) |
| C.FanSpeed1Level | Fan speed 1 threshold. Default: *25%*(AV) |
| C.FanSpeed2Level | Fan speed 2 threshold. Default: *50%*(AV) |

| Input Parameter | Description |
|---|---|
| C.FanSpeed3Level | Fan speed 3 threshold. Default: *75%*(AV) |
| C.CoolLoadRefMin | CoolLoad reference point for the MinFanSpeed. Default *0%*(AV) |
| C.FanSpdCoolMin | Minimum fan speed in the cooling mode. Default *10%*(AV) |
| C.CoolLoadRefMax | CoolLoad reference point for the MaxFanSpeed. Default *100%*(AV) |
| C.FanSpdCoolMax | Maximum fan speed in the cooling mode. Default *100%*(AV) |
| C.HeatLoadRefMin | HeatLoad reference point for the MinFanSpeed. Default *0%*(AV) |
| C.FanSpdHeatMin | Minimum fan speed in the heat mode. Default *10%*(AV) |
| C.HeatLoadRefMax | HeatLoad reference point for the MaxFanSpeed. Default *100%*(AV) |
| C.FanSpdHeatMax | Maximum fan speed in the heat mode. Default *100%*(AV) |

## Outputs

The *VariableSpeedFanCtrl (VrSdFnCt)* block output signals (hardware and logic).

| Output Parameter | Description |
|---|---|
| FanSpeedCtrl | Variable speed fan control signal. (AV) |
| FanEnableCmd | Fan start signal. (BV) |
| FanSpeedStatus | A multistate interface point (Off(1). Low(2), Med(3), High(4)). (MV) |
| EnableEquip | Signal used to enable equipment requiring fan operation. (BV) |

## Default BACnet objects

The *VariableSpeedFanCtrl (VrSdFnCt)* block related BACnet objects.

| Output Parameter | Description |
|---|---|
| FanSpeedStatus (MV 150) | A multistate interface point.<br>BACnet Multistate Value 150 output signal. |

## Block Functions

### *Variable-speed fan control algorithm*

The variable speed fan can have four different operating modes defined separately for heating and cooling actions by the C.FanHeatCtrlMode and the C.FanCoolCtrlMode. They can be defined as:

1. AlwaysOn – the actual fan speed is determined as a higher value of the C.FanEnableThresh and the value calculated from the FanHeat/CoolLoad;
2. OnDemand – the fan speed follows demand derived from the FanHeat/CoolLoad;
3. Smart – the fan speed follows the AlwaysOn scheme when the OccupancyStatus is Occupied(1) or Bypass(3) and on-demand when the OccupancyStatus is Unoccupied(2) or Standby(4);
4. Off – the fan speed is set to 0%.

### *HVACModeStatus impact*

The effective heating/cooling fan speed values are filtered by the HVACModeStatus. When the HVAC-ModeStatus is equal to:

- Auto(1), Test(8), Economy(14), Dehumid(15) – the fan speed is controlled by the higher of the heating and cooling signals;
- Heat(2), MrngWarmUp(3), MaxHeat(13) – the fan speed follows the heating load signal;
- Cool(4), NightPurge(5), PreCool(6), FreeCool(11), Ice(12) – the fan speed follows the cooling load signal;

- Off(7) – the fan speed is set to 0%;
- EmergHeat(9) – the fan speed is set to 100%;
- FanOnly(10) – the fan speed is set to C.FanDfltSpeed;

## *FanSpeedCmd input*

The FanSpeedCmd is usually connected to a room control sensor/panel and enables users to override the output FanSpeedCtrl. It is a multistate signal and can take the following values:

1. Auto – the FanSpeedCtrl output follows heating/cooling load signals;
2. Off – the FanSpeedCtrl is set to 0%;
3. Low – the FanSpeedCtrl is set to a higher value of the C.FanSpeed1Level and the C.FanEnableThresh;
4. Med – the FanSpeedCtrl is set to the C.FanSpeed2Level;
5. High – the FanSpeedCtrl is set to the C.FanSpeed3Level.

## *FanSpeedCmdPct input*

The FanSpeedCmdPct input has the same function as the FanSpeedCmd signal but holds value in per cent. When it is not null it takes precedence over the FanSpeedCmd and sets the analogue value of the FanSpeedCtrl output directly.

## *Freeze protection and HCACMode Off status*

If the HVACModeStatus is set to Off(7) the FanSpeedCtrl output value is set to 0%, unless the freeze protection mode is activated.

If the room temperature drops considerably, the HVACMode control block activates the EmergHeat(9) HVACModeStatus (for more information, refer to the HVACMode control block section). The *VariabelSpeedFanCtrl* block checks then whether the C.FanHeatCtrlMode is not equal to AlwaysOff(4) and if it is not, sets the FanSpeedCtrl output to:

- C.FanSpeed3Level if the FanSpeedCmdPct is null;
- C.FanSpeedHeatMax if the FanSpeedCmdPct is active (not null);

The freeze protection mode has the highest priority of all the overrides.

## *FanSpeedStatus output*

The FanSpeedStatus output transfers the actual FanSpeedCtrl output per cent value to a three-speed multistate value, used commonly on room control sensors/panels and on supervisory systems' graphic pages. The FanSpeedStatus uses the following switch on/off points:

Low speed:

switch-on point: the C.FanEnableThresh;

switch-off point: the C.FanEnableThresh – 5% (but not less than 2%);

Medium speed:

switch-on point: the C.FanSpeed2Level;

switch-off point: higher of the C.FanSpeed1Level and the C.FanEnableThresh;

High speed:

switch-on point: the C.FanSpeed3Level;

switch-off point: the C.FanSpeed2Level;

When the internally calculated fan control signal indicates, that the fan should be switched off (none of the stages is active), the FanSpeedStatus output will be upheld at low speed for the time defined by the C.FanOffDelay parameter. During this time the FanSpeedCtrl output is set to the C.FanEnableThresh parameter.

## *FanEnableCmd output*

The FanEnableCmd is used to operate a fan enable hardware output. It is activated together with the FanSpeedStatus output. When the FanSpeedStatus is set to Low, Medium or High speed the FanEnableCmd is also active. The C.FanOffDelay also applies to the switching-off of this output.

## *EnableEquip output*

Certain components of the FCU systems must not be turned on when the fan is not running and should be switched off before the fan is stopped (for example an electrical heater or a direct expansion cooling coil). The EnableEquip output is activated 10s after the FanEnableCmd and is switched off directly when the internally calculated fan control signal indicates, that the fan should be switched off (none of the stages is active), even though the actual fan operation is upheld for the time defined by the C.FanOffDelay parameter. It gives time for the electrical heater or DX cooler to dissipate cumulated energy.

## *EnblRetAirSample activation*

The EnblRetAirSample comes into action when an FCU uses a return temperature sensor rather than a room temperature sensor. This means that the FCU's fan must be switched on periodically to sample the room temperature even in unoccupied mode. When the algorithm decides that a sample is required it activates the EnblRetAirSample. Then the *VariabelSpeedFanCtrl* block checks whether the effective FanSpeedCmd is in the Auto(1) mode and forces the fan's speed to no less than the C.FanEnableThresh to extract room air and detect its temperature via the return air temperature sensor.

DISTECH CONTROLS™