HeatingCooling_PID

WAGO-I/O-*PRO* 32 Library elements			
Category:	Building automation		
Name:	FbPidHeatingCooling		
Type:	Function ☐	Function block ☒	Program ☐
Library name:	PidHC.lib		
Applicable to:	All programmable fieldbus controllers		
Input parameter:	**Data type:**	**Comment:**	
iInActTempIn100teldegC	INTEGER	Measured value input for room temperature	
iSetpCorrectIn100telK	INTEGER	Setpoint value correction room temp. ± 3K	
xComfortStandby	BOOL	Selection of comfort/standby operating mode (1/0)	
xNightmode	BOOL	Polling of operating mode ‘night’	
xFrostHeat	BOOL	Polling of operating mode ‘frost/heat protection’	
xDewpoint	BOOL	Polling of operating mode ‘dew point alarm’	
iSetpComfortIn100teldegC	INTEGER	Basic set point value ‘comfort mode’ Value range: 0 – 5000 [0,01^{0}C] Default setting = 2100	
iStandbyHeatingIn100telK	INTEGER	Temp. decrease standby Value range: 0 – 2000 [0,01^{0}C] Default setting = 200	
iStandbyCoolingIn100telK	INTEGER	Temp. increase standby Value range: 0 – 2000 [0,01^{0}C] Default setting = 200	
iNightHeatingIn100telK	INTEGER	Temp. decrease night Value range: 0 – 2000 [0,01^{0}C] Default setting = 400	
iNightCoolingIn100telK	INTEGER	Temp. increase night Value range: 0 – 2000 [0,01^{0}C] Default setting = 400	
iDeadzoneIn100telK	INTEGER	Dead zone between heating and cooling Value range: 10 – 1000 [0,01^{0}C] Default setting = 200	
iOffsetIn100telK	INTEGER	Measured value compensation for room temperature input Value range: -1000 – 1000 [0,01^{0}C] Default setting = 0	
dwCycleTime_10tel_s	DWORD	Scanning period of the controller [0.1s] Default setting = 600	

dwKpHeatingIn100tel	DWORD	Proportional intensification of heating [0.01] Default setting = 460
dwTiHeatingIn10tel_s	DWORD	Reset timeTi Heizen [0.1s] Default setting = 550
dwTdHeatingIn10tel_s	DWORD	Derivative time Td heating [0.1s] Default setting = 0
dwKpCoolingIn100tel	DWORD	Proportional intensification of cooling [0.01] Default setting = 460
dwTiCoolingIn10tel_s	DWORD	Reset timeTi cooling [0.1s] Default setting = 550
dwTdCoolingIn10tel_s	DWORD	Derivative time Td cooling [0.1s] Default setting = 0
Feedback value:	**Data type:**	**Comment:**
iOutIwTempIn100teldegC	INTEGER	Output of current room temperature
iOutSetpTempHeatingIn100telGrdC	INTEGER	Output of the current set value for heating
iOutSetpTempCoolingIn100telGrdC	INTEGER	Output of the current set value for cooling
xDoHeating	BOOL	Switching signal for heating valve
xDoCooling	BOOL	Switching signal for cooling valve
xConditionComfort	BOOL	Display of comfort operating mode
xConditionStandby	BOOL	Display of standby operating mode
xConditionNight	BOOL	Display of night operating mode
xConditionFrost	BOOL	Display of frost operating mode
iSetpComfortHeating	INTEGER	Outp.of current setp.value ‘comfort heating’
iSetpComfortCooling	INTEGER	Outp.of current setp.value ‘comfort cooling’
iSetpStandbyHeating	INTEGER	Outp.of current setp.val.‘standby heating’
iSetpStandbyCooling	INTEGER	Outp.of current setp.val.‘standby cooling’
iSetpNightHeating	INTEGER	Outp.of current setp.val.‘night heating’
iSetpNightCooling	INTEGER	Outp.of current setp.val.‘night cooling’
iSetpFrost	INTEGER	Output of set point value ‘frost protection’
iSetpHeat	INTEGER	Output of set point value ‘heat protection’
wAOHeating	WORD	Setting signal for valve heating Value range 0 – 32767 (0-100%)
wAOCooling	WORD	Setting signal for valve cooling Value range 0 – 32767 (0-100%)
rSetValueHeating	REAL	Setting signal for valve heating Value range 0 – 100 (0-100%)
rSetValueCooling	REAL	Setting signal for valve cooling Value range 0 – 100 (0-100%)

Graphical display:

Function description:

The function block heating / cooling allows an individual room reference temperature control while taking local influences into account. The function block can be used for heating and cooling. It compares the measured temperature value ***"iInActTempIn100teldegC"*** (actual value) with the desired heating and cooling set point values.

As an output quantity, the controller provides two respective setting signals for heating ***("wAOHeizen / rSetValueHeizen"***) and cooling ***("wAOKuehlen / rSetValueHeizen"***) of type WORD / REAL. The two outputs are only distinguishable by their value ranges. The control variable for the outputs of data type WORD is output in the value range 0 –32767 and the control variable is represented in the value range 0 – 100 for data type REAL.

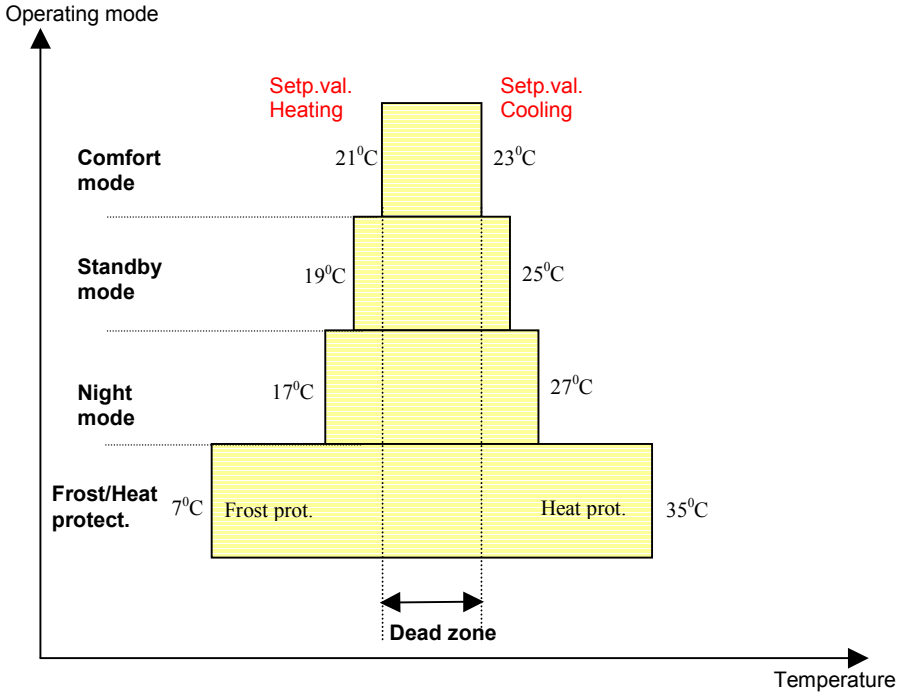
The controller detects four operating modes (conditions) to each of which is assigned its own set point value. The ***"iSetpComfort100teldegC"*** is used as a set point value. All other set point values refer to the basic set point value and provoke each a set point value increase or set point value decrease by a parameterized value. Th basic set point value can be infinitely shifted by ± 3 K via the ***"iSetpCorrectIn100telK"*** input. The active operating mode (comfort, standby, night, frost protection, dew point alarm) is determined via the communication objects ***("xComfortStandby", "xNightmode", "xFrostHeat", "xDewpoint"***). The currently selected operating mode is viasualized ***via "xConditionComfort", "xConditionStandby", "xConditionNight", "xConditionFrost"***. A dead zone ***("iDeadZoneIn100telK"***) has to be parameterized between the operating modes heating and cooling. The selected size of this dead zone must not be too small in order to avoid a permanent switch-over between heating and cooling. The room temperature measured and, if applicable, compensated with the ***"iOffsetIn100telK"*** parameter can be sent via the output object ***"iOutIwTempIn100teldegC"***.

The max. deviation in relation to the set point temperature is entered as a default value via parameter ***" iHystIn100telK"***. A small hysteresis provokes a frequent switching of the valve voltage, but small set point value differences. A large hysteresis causes large deviations from the set point value, but leads to rare switching. If the function module is used for cooling purposes, another input object is required. The name of this object is ***"xDewpoint"***. If a dew point alarm is signalled on this object, the cooling / heating system switches off immediately.

The parameter **"dwAbtastzeit"** determines the scanning period of the controller.
The controller is set for heating via the parameters ***"dwKpHeizenIn100stel", "dwTiHeizenIn10tel_s" and "dwTdHeizenIn10tel_s".***
The controller is set for cooling via the parameters ***"dwKpKuehlenIn100stel", "dwTiKuehlenIn10tel_s" and "dwTdKuehlenIn10tel_s".***
Whereby the individual parameters have the following meaning:

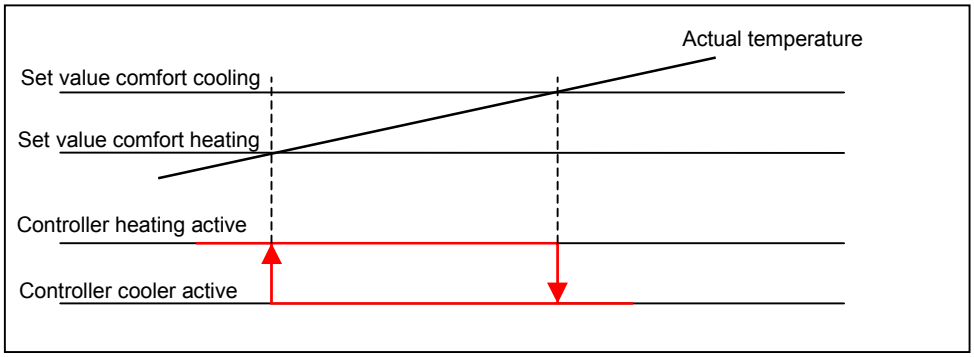
- dwKp = proportional intensification. The proportional intensification marks the P-part of the controller and indicates how "hard" the controller reacts towards changes in temperature.
- dwTi = reset time. The reset time marks the I-part of the controller.
- dwTd = derivative time. The derivative time marks the D-part of the controller.

The function module has ten monitor outputs: ***"iOutSetpHeatingIn100telGrdC", "iOutSetpCoolingIn100telGrdC", "iSetpComfortHeating", "iSetpComfortCooling "iSetpStandbyHeating", "iSetpStandbyCooling", "iSetpNightHeating", "SetpNightCooling", "iSetpFrost" and "iSetpHeat"***. The current set point values of the individual operating modes are put out via these outputs.



Operating mode	Setpoint value Heating	Setpoint value Cooling
Comfort mode	**Basic setpoint value 21 ^{0}C**	Basic set point value +dead zone 2 K
Standby mode	Basic set point value - Lowering, standby mode	Basic set point value +dead zone +raising, standby mode
Night mode	Basic set point value - Lowering, night mode	Basic set point value +dead zone +raising, night mode
Frost/heat protection	Set point value, frost protection 7 ^{0}C	Set point value, heat protection 35 ^{0}C
Dew point alarm	Setting value 0 (heating off)	Setting value 0 (cooling off)

The outputs ***"xDoHeating"*** and ***"xDoCooling"*** show which mode (heating or cooling) is active. If the setting signal for heating and cooling is 0%, then the two outputs xDoHeating and xDoCooling also have the signal "FALSE". The switching between heating and cooling takes place automatically (see diagram below). The controller is either in the heating mode or in the cooling mode. The mode that is currently not active is switched to 0%.



Important!
The function module FbPidHeatingCooling can only be used in connection with the library Control.lib.

Note:

- The D-part is set to zero with most of the room heating controllers because a PI controller has sufficient precision and is easier to set.
- This function block uses some residual variables having a **VAR_RETAIN** declaration.