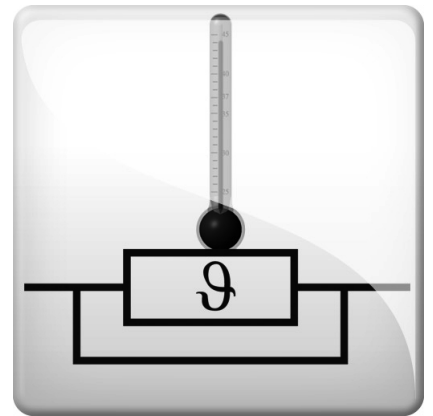


Technology module



Temperature Control

Reference manual

EN

Contents

1	About this documentation	3
1.1	Document history	5
1.2	Conventions used	6
1.3	Layout of the safety instructions	7
2	Safety instructions	8
3	Overview of the technology module functions	10
3.1	Functional description	10
3.2	Overview of the functions	10
4	L_TP2P_TemperatureControlBase function block	11
4.1	Inputs and outputs	11
4.2	Inputs	12
4.3	Outputs	13
4.4	Parameter	15
4.5	Persistent data	19
5	State Machine	20
5.1	State machine	20
6	Signal flow diagrams	22
6.1	Structure of the signal flow	24
6.2	Structure of the access points	25
6.3	Use of the access points	27
7	Basic settings	28
7.1	Maximum temperature limit	28
7.2	Task cycle time	28
7.3	Minimum external wiring for the use of the PWM module	29
7.4	Minimum external wiring for the use of the DIFF module	30
8	Description of the technology functions	31
8.1	Parameter identification	31
8.1.1	Activation	31
8.1.2	Procedure after activating the parameter identification	32
8.1.3	Signal characteristic	36
8.1.4	Persistent data	37
8.2	Temperature control	38
8.2.1	Activation	38
8.2.2	Signal characteristic	39
8.2.3	Temperature control with parameter switching	40
8.2.4	Temperature control with differential actuator	41
8.3	Control with constant correcting variable	42
8.3.1	Activation	42
8.3.2	Signal characteristic	43
	Index	44
	Your opinion is important to us	46

1 About this documentation

This documentation ...

- contains detailed information on the functionalities of the "Electrical Shaft Position" technology module;
- is part of the "Controller-based Automation" manual collection. It consists of the following sets of documentation:

Documentation type	Subject
Product catalogue	Controller-based Automation (system overview, sample topologies) Lenze Controller (product information, technical data)
System manuals	Visualisation (system overview/sample topologies)
Communication manuals Online helps	Bus systems • Controller-based Automation EtherCAT® • Controller-based Automation CANopen® • Controller-based Automation PROFIBUS® • Controller-based Automation PROFINET®
Reference manuals Online helps	Lenze Controllers: • Controller 3200 C • Controller c300 • Controller p300 • Controller p500
Software manuals Online helps	Lenze Engineering Tools: • »PLC Designer« (programming) • »Engineer« (parameter setting, configuration, diagnostics) • »VisiWinNET® Smart« (visualisation) • »Backup & Restore« (data backup, recovery, update)

More technical documentation for Lenze components

Further information on Lenze products which can be used in conjunction with Controller-based Automation can be found in the following sets of documentation:

Planning / configuration / technical data	
☐	Product catalogues • Controller-based Automation • Controllers • Inverter Drives/Servo Drives
Mounting and wiring	
☒	Mounting instructions • Controllers • Communication cards (MC-xxx) • I/O system 1000 (EPM-Sxxx) • Inverter Drives/Servo Drives • Communication modules
☐	Hardware manuals • Inverter Drives/Servo Drives
Parameter setting / configuration / commissioning	
☐	Online help/reference manuals • Controllers • Inverter Drives/Servo Drives • I/O system 1000 (EPM-Sxxx)
☐	Online help/communication manuals • Bus systems • Communication modules
Sample applications and templates	
☐	Online help / software and reference manuals • i700 application sample • Application Samples 8400/9400 • FAST Application Template Lenze/PackML • FAST technology modules

Symbols:

-  Printed documentation
- ☐ PDF file / online help in the Lenze engineering tool



Tip!

Current documentation and software updates with regard to Lenze products can be found in the download area at:

www.lenze.com

Target group

This documentation is intended for all persons who plan, program and commission a Lenze automation system on the basis of the Lenze FAST Application Software.

1 About this documentation

1.1 Document history

1.1 Document history

Version			Description
7.0	04/2021	TD06	Error corrections & supplements
6.0	06/2020	TD06	Error corrections & supplements
5.0	12/2019	TD06	Error corrections & supplements
4.0	06/2019	TD29	Double parameter description removed Extended: ‣ Parameter (15) IrDiffCtrlSignalTimeMin, IrDiffCtrlBreakTimeMin ‣ Temperature control with differential actuator (41)
3.0	07/2018	TD29	New: ‣ Minimum external wiring for the use of the PWM module (29) ‣ Minimum external wiring for the use of the DIFF module (30) ‣ Temperature control with parameter switching (40) ‣ Temperature control with differential actuator (41) Extended: ‣ Parameter (15)
2.2	05/2017	TD17	Content structure has been changed. New: ‣ Outputs (13) xDiffPositiveOut and xDiffNegativeOut ‣ Parameter (15) IrDiffCtrlActuatingTime ‣ Correcting variable conditioning (23) ‣ Minimum external wiring for the use of the DIFF module (30)
2.1	04/2016	TD17	General revisions
2.0	10/2015	TD17	• General editorial revision • Modularisation of the contents for the »PLC Designer« online help
1.0	06/2015	TD00	First edition

1 About this documentation

1.2 Conventions used

This documentation uses the following conventions to distinguish between different types of information:

Type of information	Highlighting	Examples/notes
Spelling of numbers		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Program name	» «	»PLC Designer« ...
Variable names	<i>italics</i>	By setting <i>bEnable</i> to TRUE...
Function blocks	bold	The L_MC1P_AxisBasicControl function block ...
Function libraries		The L_TT1P_TechnologyModules function library ...
Source code	Font "Courier new"	... dwNumerator := 1; dwDenominator := 1; ...
Icons		
Page reference	 6	Reference to further information: Page number in PDF file.

Variable names

The conventions used by Lenze for the variable names of Lenze system blocks, function blocks, and functions are based on the "Hungarian Notation". This notation makes it possible to identify the most important properties (e.g. the data type) of the corresponding variable by means of its name, e.g. *xAxisEnabled*.

1.3

Layout of the safety instructions

** DANGER!**

Indicates an extremely hazardous situation. Failure to comply with this instruction will result in severe irreparable injury and even death.

** WARNING!**

Indicates an extremely hazardous situation. Failure to comply with this instruction may result in severe irreparable injury and even death.

** CAUTION!**

Indicates a hazardous situation. Failure to comply with this instruction may result in slight to medium injury.

NOTICE

Indicates a material hazard. Failure to comply with this instruction may result in material damage.

Application notes

Pictograph	Signal word	Meaning
	Tip!	Useful tip for easy handling
		Reference to another document

2 Safety instructions

Please observe the safety instructions in this documentation when you want to commission an automation system or a plant with a Lenze Controller.



The device documentation contains safety instructions which must be observed!

Read the documentation supplied with the components of the automation system carefully before you start commissioning the Controller and the connected devices.

DANGER!

High electrical voltage

Injury to persons caused by dangerous electrical voltage

Possible consequences

Death or severe injuries

Protective measures

Switch off the voltage supply before working on the components of the automation system.

After switching off the voltage supply, do not touch live device parts and power terminals immediately because capacitors may be charged.

Observe the corresponding information plates on the device.

DANGER!

Injury to persons

Risk of injury is caused by ...

- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Possible consequences

Death or severe injuries

Protective measures

- If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
 - During commissioning, maintain an adequate safety distance to the motor or the machine parts driven by the motor.
-

NOTICE

Damage or destruction of machine parts

Damage or destruction of machine parts can be caused by ...

- Short circuit or static discharges (ESD);
- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Protective measures

- Always switch off the voltage supply before working on the components of the automation system.
 - Do not touch electronic components and contacts unless ESD measures were taken beforehand.
 - If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
-

3 Overview of the technology module functions

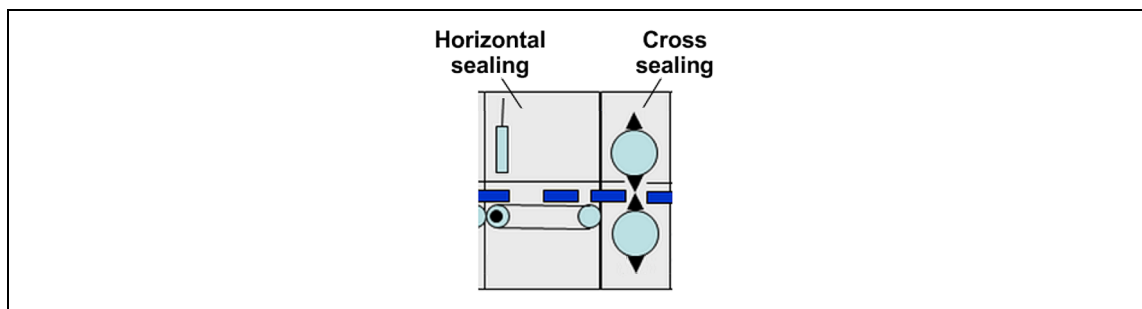
3.1 Functional description

3 Overview of the technology module functions

3.1 Functional description

The "Temperature Control" technology module can be used in applications in which the primary task is the control of the temperature of a (partial) system that is equipped with a heating element and a thermal sensor.

Typical applications in packaging machines for example are temperature controls for sealing knives or sealing bars which seal and/or separate packaging material by the effect of heat. Another typical application is found in extruders in the case of which material such as synthetic granules is sealed and compressed.



[3-1] Sealing knife and sealing bar in packaging machines

3.2 Overview of the functions

Besides the basic functions, such as switching on and off via the input *xEnable*, the technology module offers the following functionalities:

▶ [Parameter identification](#) (📘 31)

In this operating mode, the optimal values for the controller parameters of the heat section are identified.

▶ [Temperature control](#) (📘 38)

In this operating mode, the heat section is controlled with the identified controller parameters. Before temperature control, the parameter identification has to be carried out once.

▶ [Control with constant correcting variable](#) (📘 42)

In this operating mode, the heat section is controlled with a constant correcting variable defined in the parameters.

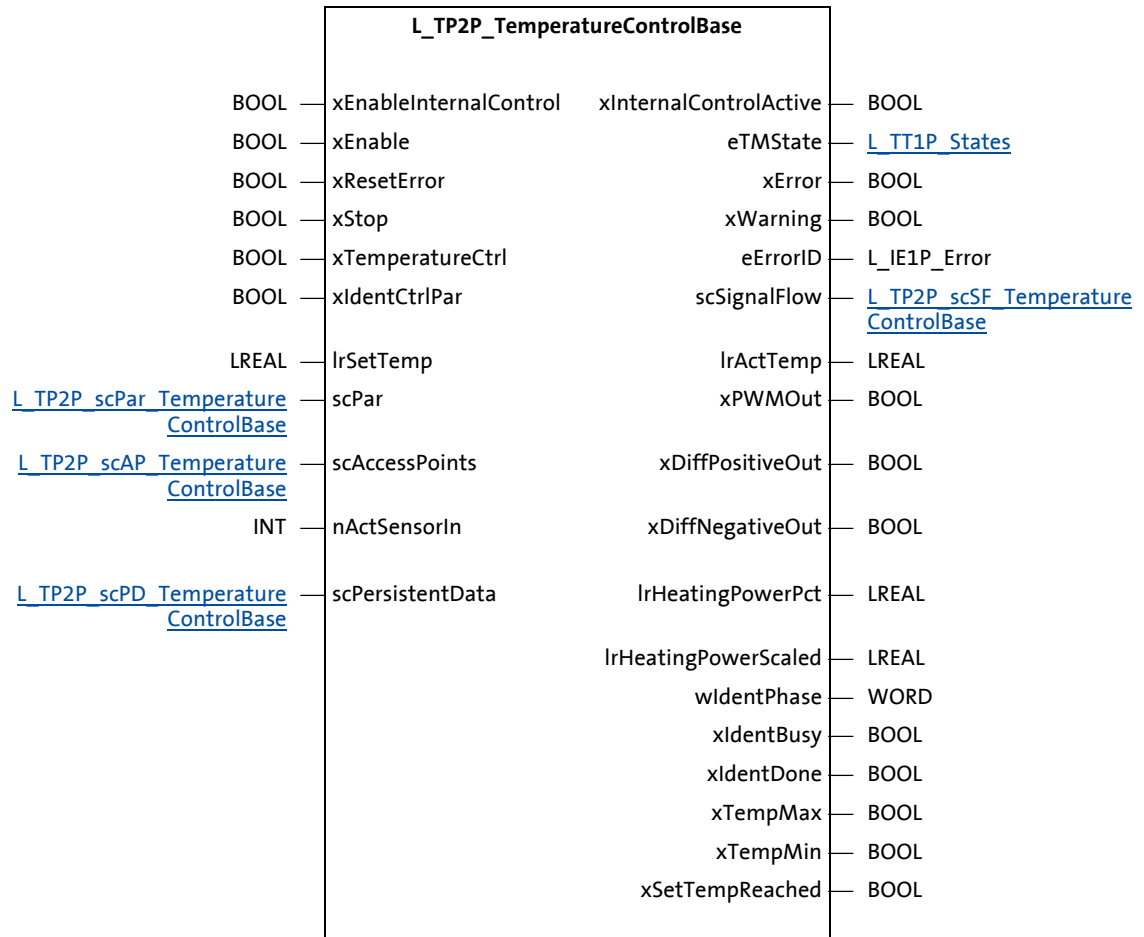
This is required for maintaining an emergency operation when the thermal sensor or its evaluation unit is defective.

All temperature monitoring modes are deactivated in this operating mode!

4 L_TP2P_TemperatureControlBase function block

The figure shows the inputs and outputs of the function block.

The base version offers the full functionality of the technology module.



4.1 Inputs and outputs

Designator	Description
Data type	
scPersistentData L_TP2P_scPD_TemperatureControlBase	Persistent data (19) contain the controller characteristic values calculated in the course of the parameter identification process (xIdentCtrlPar input = TRUE).

4.2

Inputs

Designator	Data type	Description
xEnableInternalControl	BOOL	TRUE In the visualisation, the internal control of the axis can be selected via the "Internal Control" axis.
xEnable	BOOL	Execution of the function block
		TRUE The function block is executed.
		FALSE The function block is not executed.
xResetError	BOOL	FALSE/T TRUE Resetting errors of the "Temperature Control" technology module.
xStop	BOOL	TRUE The technology module is stopped, i.e. the temperature control is stopped. - The output <i>xPWMOut</i> is permanently set to FALSE. - The state changes to "STOP". - The "STOP" state is left when the input <i>xStop</i> = FALSE is set. - The input is also switched through with "Internal Control" from the application.
xTemperatureCtrl	BOOL	TRUE Activate the " Temperature control (□ 38) " or " Control with constant correcting variable (□ 42) " operating mode. The operating mode is selected via the <i>eModeTempCtrl</i> parameter of the L_TP2P_scPar_TemperatureControlBase (□ 15) parameter structure
xIdentCtrlPar	BOOL	TRUE Activate the " Parameter identification (□ 31) " operating mode. The values to be used are selected via the <i>xIdentAuto</i> parameter of the L_TP2P_scPar_TemperatureControlBase (□ 15) parameter structure
lrSetTemp	LREAL	Temperature setpoint for the "Temperature control" operating mode - Unit: °C or °F - Depending on the temperature unit set in the <i>xTempUnitIsCelsius</i> parameter of the L_TP2P_scPar_TemperatureControlBase (□ 15) parameter structure.
scPar L_TP2P_scPar_TemperatureControlBase		The parameter structure contains the parameters of the technology module.
scAccessPoints L_TP2P_scAP_TemperatureControlBase		Structure of the access points
nActSensorIn	INT	Input for coupling the thermal sensor (usually PT100 with interface connection via EPM module)

4.3

Outputs

Designator Data type	Description	
xInternalControlActive BOOL	TRUE	The internal control of the technology module is activated via the visualisation. (xEnableInternalControl input = TRUE)
eTMState L TT1P_States	Current state of the technology module ► State Machine (□ 20)	
	210	CONTROL
	211	REGULATION
	212	IDENTIFICATION
xError BOOL	TRUE	There is an error in the technology module.
xWarning BOOL	TRUE	There is a warning in the technology module.
eErrorID L_IE1P_Error	ID of the error or warning message if <i>xError</i> = TRUE or <i>xWarning</i> = TRUE. "FAST technology modules" reference manual: Here you can find information on error or warning messages.	
scSignalFlow L_TP2P_scSF_TemperatureControlBase	Structure of the signal flow ► Signal flow diagrams (□ 22)	
IrActTemp LREAL	Actual temperature value • Unit: °C or °F Depending on the temperature unit set in the <i>xTempUnitIsCelsius</i> parameter of the L_TP2P_scPar_TemperatureControlBase (□ 15) parameter structure.	
xPWMOut BOOL	TRUE	Output signal of the internal PWM module for triggering a (solid state) relay
xDiffPositiveOut BOOL	Positive output signal for actuators (mixing valves) with a differential control	
	TRUE	Open mixing valve / increase heating power
xDiffNegativeOut BOOL	Negative output signal for actuators (mixing valves) with a differential control	
	TRUE	Close mixing valve / reduce heating power
IrHeatingPowerPct LREAL	Current correcting variable for the heating power • Unit: %	
IrHeatingPowerScaled LREAL	Current correcting variable for the heating power • Value range: 0.0 ... 1.0 (1.0 = 100 %)	
wIdentPhase LREAL	Display of the active phase during the parameter identification Possible values:	
	0	Parameter identification is inactive
	1	Parameter identification phase 1 active
	2	Parameter identification phase 2 active
	3	Parameter identification phase 3 active
	4	Parameter identification phase 4 active
	5	Parameter identification is completed
xIdentBusy BOOL	TRUE	Parameter identification is active
xIdentDone BOOL	TRUE	The parameter identification has been completed successfully.

Designator	Data type	Description	
xTempMax	BOOL	TRUE	Warning signal "Overtemperature" The current actual temperature exceeds the setpoint temperature by more than the value set in the IrTempMaxWindow (parameter structure L_TP2P_scPar_TemperatureControlBase (15)) parameter.
xTempMin	BOOL	TRUE	Warning signal "Undertemperature" The current actual temperature falls below the setpoint temperature by more than the value set in the IrTempMinWindow parameter (L_TP2P_scPar_TemperatureControlBase (15) parameter structure).
xSetTempReached	BOOL	TRUE	Information signal "Setpoint temperature reached" - The temperature setpoint (IrSetTemp input) has been reached. - The deviation between the setpoint temperature and the actual temperature does <u>not</u> exceed the tolerance set in the IrSetTempReachedTolerance parameter (L_TP2P_scPar_TemperatureControlBase (15) parameter structure).

4.4

Parameter

L_TP2P_scPar_TemperatureControlBase

The **L_TP2P_scPar_TemperatureControlBase** structure contains the parameters of the technology module.

Designator	Data type	Description
IrSensorValueGain	LREAL	Gain factor for the sensor signal. Set this value so that the sensor used provides a correct mapping to the temperature unit °C. • Unit: - • Initial value: 0.1
IrSensorValueOffset	LREAL	Offset factor for the scaled sensor signal. Set this value so that the sensor used and the configured gain factor provide a correct zero point mapping (absolute position) to the temperature unit °C. • Unit: - • Initial value: 0.0
xTempUnitIsCelsius	BOOL	Unit of temperature values • Initial value: TRUE
		TRUE All temperature values (setpoint / actual value / configuration) are based on the Celsius [°C] unit.
		FALSE All temperature values (setpoint / actual value / configuration) are based on the Fahrenheit [°F] unit.
IrTempMaxCritical Absolute	LREAL	Absolute temperature limit for the system. If this value is reached, the correcting variable is set to 0 % and an error is reported. • Unit: °C (°F) • Initial value: 120.0
IrSetTempReached Tolerance	LREAL	As long as the deviation between the target and actual temperature does not exceed the value configured here, the status signal <i>xSetTempReached</i> = TRUE is set. • Unit: °C (°F) • Initial value: 1.0
IrTempMinWindow	LREAL	If the actual value exceeds the setpoint value by more than the difference specified here, the warning signal is set to <i>xTempMin</i> = TRUE. • Unit: °C (°F) • Initial value: 5.0
IrTempMinHysteresis	LREAL	Hysteresis for the <i>xTempMin</i> warning signal. • Unit: °C (°F) • Initial value: 1.0
IrTempMaxWindow	LREAL	If the actual value exceeds the setpoint value by more than the difference specified here, the warning signal is set to <i>xTempMax</i> = TRUE. • Unit: °C (°F) • Initial value: 5.0
IrTempMaxHysteresis	LREAL	Hysteresis for the <i>xTempMax</i> warning signal. • Unit: °C (°F) • Initial value: 1.0
IrSensorMonitoring ValueMin	LREAL	If the sensor delivers a resulting temperature value of less than this parameter value, this is recognised as a probe error and an error is reported. • Value -1000.0 deactivates this monitoring. • Unit: °C (°F) • Initial value: -1000.0
IrSensorMonitoring ValueMax	LREAL	If the sensor provides a resulting temperature value greater than this parameter value, this is recognised as a probe error and an error is reported. • Value 1000.0 deactivates this monitoring. • Unit: °C (°F) • Initial value: 1000.0

Designator	Data type	Description
IrSensorMonitoringChangeMax	LREAL	If the change of the resulting temperature value within a cycle is greater than the value specified here, this is recognised as a probe error and an error is reported. - Value 0.0 deactivates this monitoring. - Unit: °C (°F) - Initial value: 0.0
xCtrlIntegralClamping	BOOL	Controller behaviour Initial value: TRUE
		TRUE As long as the temperature controller is within the correcting variable limitation, the added-up I component is frozen.
IrControlValueMinPct	LREAL	Minimum value for the correcting variable limitation of the temperature controller. - Unit: % - Initial value: 0.0
IrControlValueMaxPct	LREAL	Maximum value for the correcting variable limitation of the temperature controller. - Unit: % - Initial value: 100.0
IrPWMPeriod	LREAL	Period of the PWM signal generated in the TM. - Unit: s - Initial value: 1.0
IrDiffCtrlActuatingTime	LREAL	Runtime of a connected differential actuator. The runtime is the time period required by the control valve from full closing to full opening with permanent activation. - Unit: s - Initial value: 60.0 ► Temperature control with differential actuator (41)
IrDiffCtrlSignalTimeMin	LREAL	Minimum pulse duration at the <i>xDiffPositiveOut</i> and <i>xDiffNegativeOut</i> outputs; for reducing the strain on the electromechanical valve control components. - Unit: s - Initial value: 0.3 ► Temperature control with differential actuator (41)
IrDiffCtrlBreakTimeMin	LREAL	Minimum break duration at the <i>xDiffPositiveOut</i> and <i>xDiffNegativeOut</i> outputs for reducing the strain on the electromechanical valve components. This break duration is also complied with when switching over from positive to negative. - Unit: s - Initial value: 0.3 ► Temperature control with differential actuator (41)
eModeTempCtrl	ENUM	CONTROL setting: A constant correcting variable for the heat section is set, all temperature monitoring is deactivated, cf. parameter IrControlValueFixedPct REGULATION setting: The heat section is controlled and operated with the controller. - Unit: - - Initial value: REGULATION
IrControlValueFixedPct	LREAL	Constant correcting variable for the "CONTROL" operating mode. - Unit: % - Initial value: 10.0
xIdentAuto	BOOL	Sequence of the identification process Initial value: TRUE
		TRUE For the identification process, standard values and automatically calculated values are used.
		FALSE For the identification process, the values stored in the following parameters are used.
IrIdentStep1TempChangeMax	LREAL	Maximum permissible temperature change so that reaching the stationary (ambient) temperature is detected in identification step 1 and step 1 can be terminated. - If the parameter <i>xIdentAuto</i> = TRUE, this parameter is not used. - Unit: °C/min (°F/min) - Initial value: 0.5

Designator	Data type	Description
IrIdentStep2ControlStepPct	LREAL	Initial correcting variable injection in identification step 2. - If the parameter <i>xIdentAuto</i> = TRUE, this parameter is not used. - Unit: % - Initial value: 4
IrIdentStep2TempChangeMax	LREAL	Maximum permissible temperature change so that reaching the stationary temperature is detected in identification step 2 and step 2 can be terminated. - If the parameter <i>xIdentAuto</i> = TRUE, this parameter is not used. - Unit: °C/min (°F/min) - Initial value: 0.5
IrIdentStep2TempDiffMin	LREAL	Minimum value of the temperature increase to be reached in identification step 2 before switching over to identification step 3. - If the parameter <i>xIdentAuto</i> = TRUE, this parameter is not used. - Unit: °C/min (°F/min) - Initial value: 10.0
IrIdentStep2WaitTimeMin	LREAL	If the stationary temperature (<i>IrIdentStep2TempChangeMax</i> parameter) has been reached after this time has elapsed but not the minimum temperature increase from the <i>IrIdentStep2TempDiffMin</i> parameter, the correcting variable will be doubled and the time will be restarted. - If the parameter <i>xIdentAuto</i> = TRUE, this parameter is not used. - Unit: min - Initial value: 15
IrIdentStep2ControlValueMaxPct	LREAL	If the correcting variable exceeds this value without the minimum temperature increase configured in the <i>IrIdentStep2TempDiffMin</i> parameter having been reached, the identification ends with an error. - If the parameter <i>xIdentAuto</i> = TRUE, this parameter is not used. - Unit: % - Initial value: 50
IrIdentStep3ControlStepPct	LREAL	Value of the correcting variable step for accepting the heat section step response in identification step 3. If the value is "-1", the correcting variable step for identification step 3 is automatically calculated from the heating process of identification step 2. - Unit: % - Initial value: 20
IrIdentStep3TempDiffMin	LREAL	Minimum value of the temperature increase to be reached in identification step 3 in order for the detected parameters to be regarded as valid. - Unit: °C/min (°F/min) - Initial value: 20.0
IrIdentStep3RegressionTime	LREAL	A linear regression is used to determine the slope of the heating curve. This parameter defines the time length of the regression interval. If the parameter <i>xIdentAuto</i> = TRUE, this parameter is not used. The value must be at least one hundred times the task cycle time. If this is not fulfilled, error no. 17171 (TaskCycleRegressionTimeMismatch) is triggered when the parameter identification is started. - Unit: s - Initial value: 2.0
xActivateParameterSwitchung	BOOL	Control process Initial value: FALSE
		TRUE Control process with parameter switching active
		FALSE Control process with parameter switching inactive
IrDeviationConstraintA	LREAL	When this system deviation limit value is not reached, the parameter set is activated for optimal disturbance behaviour for the control process with parameter switching. - This parameter is only effective for <i>xActivateParameterSwitchung</i> = TRUE - Unit: °C [°F] - Initial value: 1.0

Designator	Data type	Description
IrDeviationConstraintB	LREAL	When this system deviation limit value is exceeded, the parameter set is activated for optimal guidance behaviour for the control process with parameter switching. • This parameter is only effective for <i>xActivateParameterSwitching</i> = TRUE • Unit: °C [°F] • Initial value: 3.0
IrGainFactorDisturbanceControl	LREAL	Assessment factor for controller gain relating to K_{crit} (for Ziegler-Nichols) for the parameter set for optimal disturbance behaviour for the control process with parameter switching. • This parameter is only effective for <i>xActivateParameterSwitching</i> = TRUE • Unit: - • Initial value: 2.0
IrFilterTimeFactorDisturbanceControl	LREAL	Assessment factor for the time constant of the setpoint filter relating to dead time of the control section for the parameter set for optimal disturbance behaviour for the control process with parameter switching. • This parameter is only effective for <i>xActivateParameterSwitching</i> = TRUE • Unit: - • Initial value: 6.0
IrGainFactorReferenceControl	LREAL	Assessment factor for controller gain relating to K_{crit} (for Ziegler-Nichols) for the parameter set for optimal guidance behaviour for the control process with parameter switching. • This parameter is only effective for <i>xActivateParameterSwitching</i> = TRUE • Unit: - • Initial value: 0.5
IrFilterTimeFactorReferenceControl	LREAL	Assessment factor for time constant of the setpoint filter relating to dead time of the control section for the parameter set for optimal guidance behaviour for the control process with parameter switching. • This parameter is only effective for <i>xActivateParameterSwitching</i> = TRUE • Unit: - • Initial value: 3.0

4.5

Persistent data

L_TP2P_scPD_TemperatureControlBase

The **L_TP2P_scPD_TemperatureControlBase** parameter structure contains the controller characteristic values calculated in the course of the parameter identification process (*xIdentCtrlPar* input = TRUE).

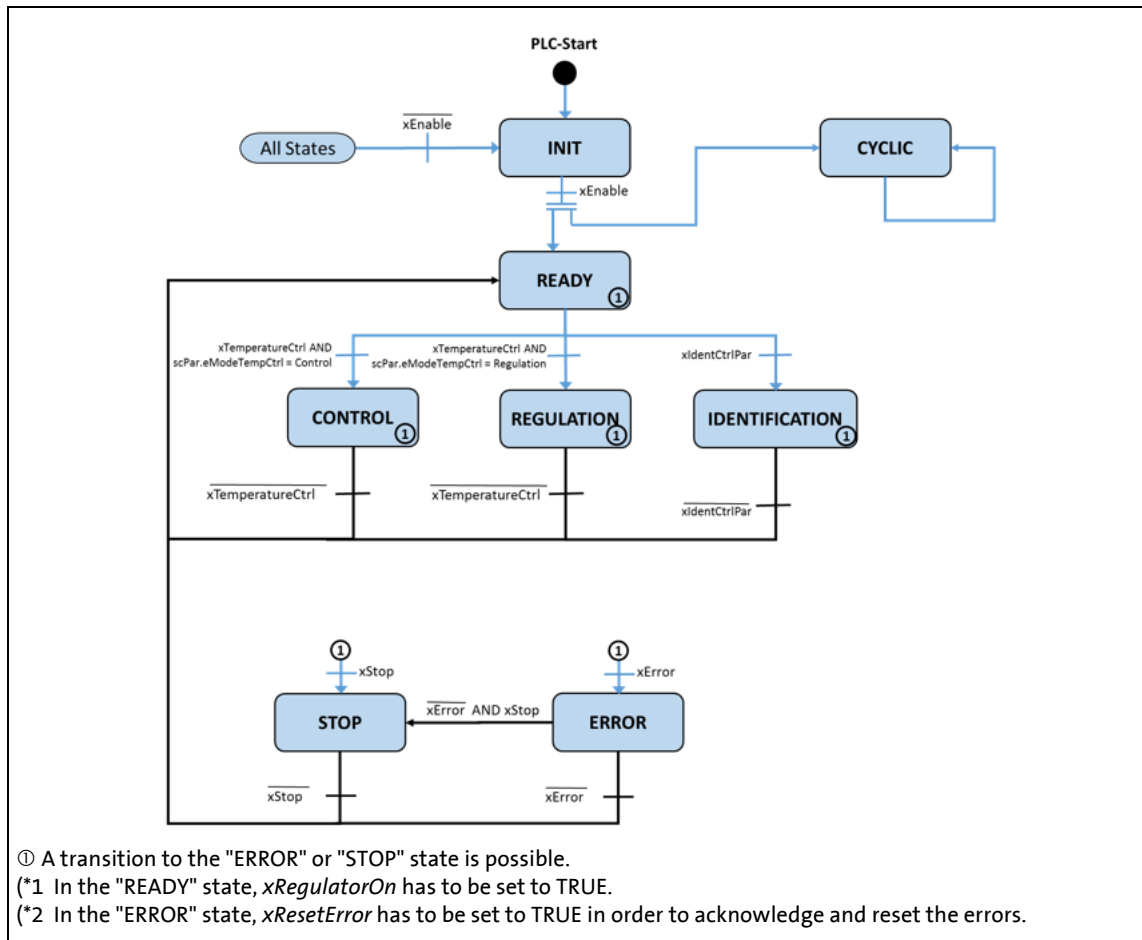
Designator Data type	Description
IrTempSetFilterTime LREAL	PT1 filter time constant for the temperature setpoint With the value '0', filtering is deactivated. - Unit: s - Initial value: 30.0
IrTempActFilterTime LREAL	PT1 filter time constant for the actual temperature value With the value '0', filtering is deactivated. - Unit: s - Initial value: 5.0 Example: A task cycle time of 10 ms and the common factor of 500 result in a filter time constant of 5 seconds.
IrTempCtrlGain BOOL	P parameter of the PID controller - Unit: %/°C / %/°F - Initial value: 0.0
IrTempCtrlResetTime LREAL	Reset time (I parameter) of the PID controller - Unit: s - Initial value: 100.0
IrTempCtrlRateTime LREAL	Rate time (D parameter) of the PID controller - Unit: s - Initial value: 100.0
dwIdentVersionInfo DWORD	Version identifier of the parameters detected by the parameter identification - This serves to detect whether valid parameters are available (dwIdentVersionInfo > 0). - Moreover, an external persistence mechanism can detect whether new parameters have been identified since the value of dwIdentVersionInfo is increased by 1 each time. - Initial value: 0

5 State Machine

5.1 State machine

5 State Machine

5.1 State machine



[5-1] State machine of the technology module

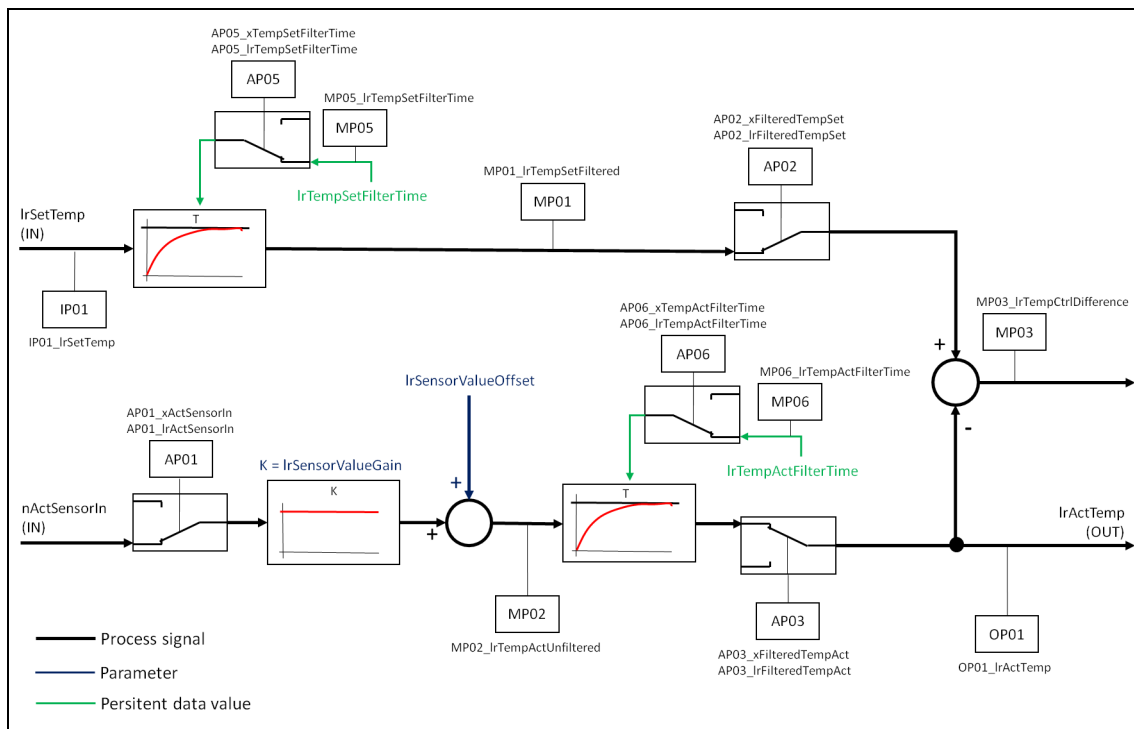
States of the eTMState output (L_TT1P_States)

No.	L_TT1P_States	Description
1	INIT	Initialisation of the technology module active.
2	READY	Technology module ready for operation.
210	CONTROL	In this operating mode, the heat section is controlled with a constant correcting variable defined in the parameters.
211	REGULATION	In this operating mode, the heat section is controlled with the identified controller parameters.
212	IDENTIFICATION	In this operating mode, the optimal values for the controller parameters of the heat section are identified.
996	STOP	Stop/Halt active.
998	SERVICE	The technology module is in service mode. The internal L_MC1P_AxisBasicControl function block is controlled via the <i>scCtrlABC</i> input structure. The status of the function block can be viewed via the <i>scStatusABC</i> output structure.

No.	L_TT1P_States	Description
999	ERROR	Error state
1000	SYSTEMFAULT	System fault

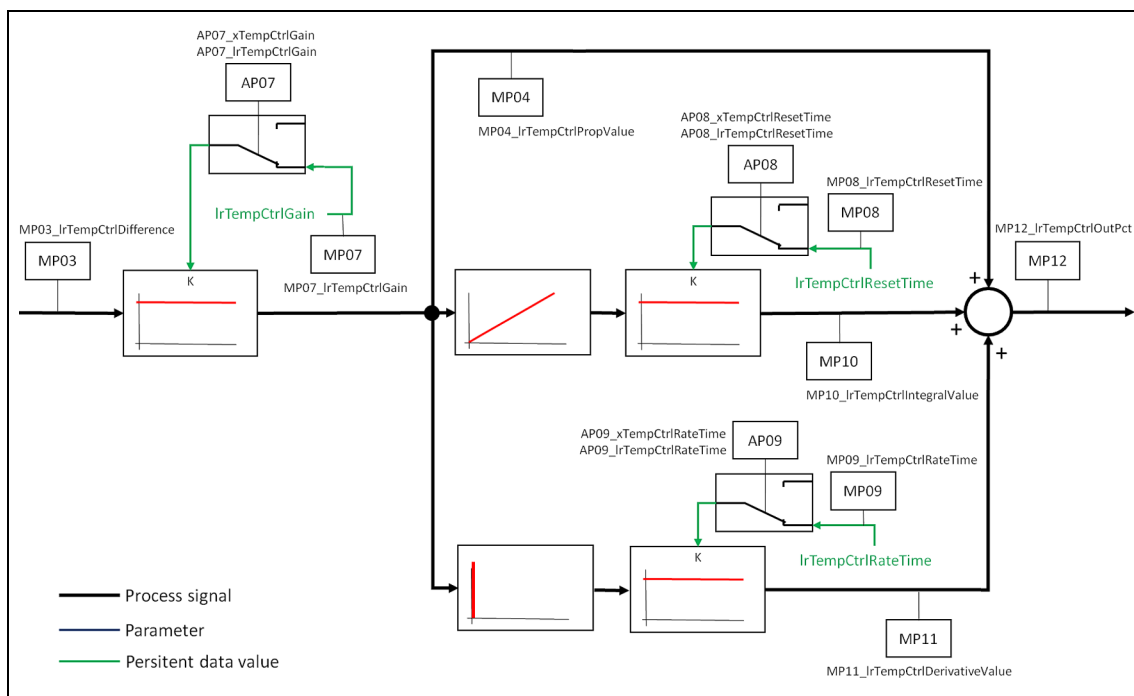
6 Signal flow diagrams

Setpoint/actual value conditioning



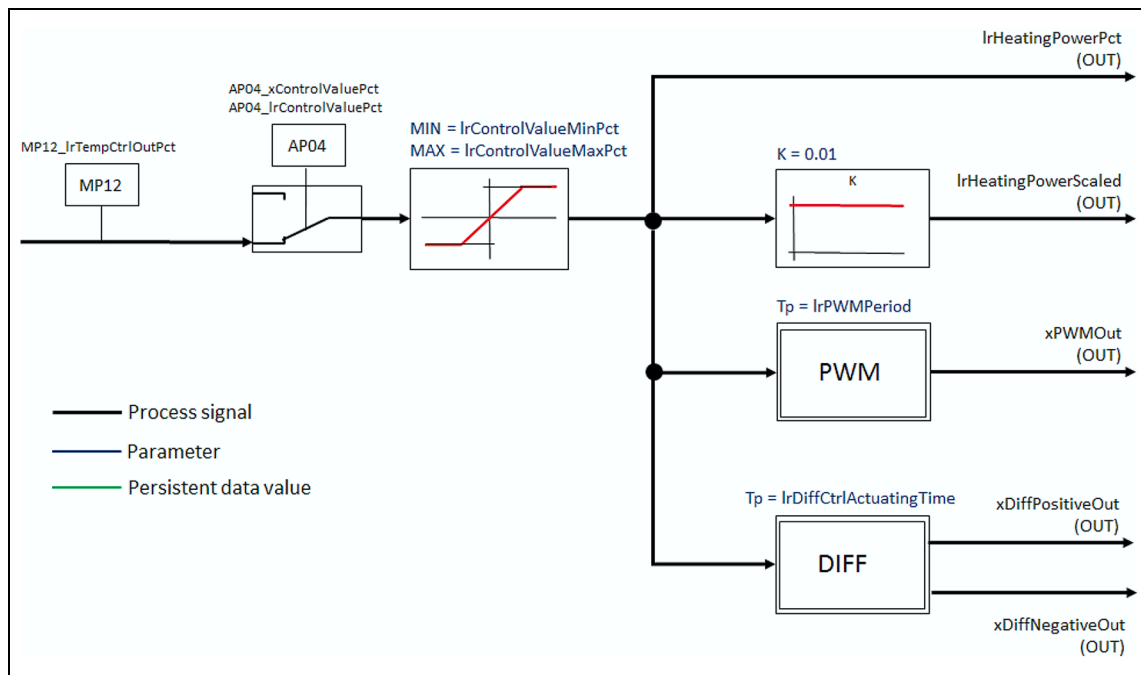
[6-1] Signal flow diagram: Setpoint/actual value conditioning

Thermostat



[6-2] Signal flow diagram: Thermostat

Correcting variable conditioning



[6-3] Signal flow diagram: Correcting variable conditioning

6.1 Structure of the signal flow

L_TP2P_scSF_TemperatureControlBase

The contents of the **L_TP2P_scSF_TemperatureControlBase** structure are read-only and offer a practical diagnostics option within the signal flow ([Signal flow diagrams](#) (□ 22)).

Designator	Data type	Description
IP01_IrSetTemp	LREAL	Temperature setpoint for the "temperature control" operating mode (eModeTempCtrl parameter = REGULATION) • Unit: °C or °F
MP01_IrTempSetFiltered	LREAL	Filtered temperature setpoint • Unit: °C or °F
MP02_IrTempActUnfiltered	LREAL	Unfiltered actual temperature value • Unit: °C or °F
MP03_IrTempCtrlDifference	LREAL	System deviation between temperature setpoint and actual value • Unit: °C or °F
MP04_IrTempCtrlPropValue	LREAL	Process value of the proportional component (P) of the controller
MP05_IrTempSetFilterTime	LREAL	PT1 filter time constant for the temperature setpoint With the value '0', filtering is deactivated. • Unit: s
MP06_IrTempActFilterTime	LREAL	PT1 filter time constant for the actual temperature value With the value '0', filtering is deactivated. • Unit: s Example: A task cycle time of 10 ms and the common factor of 500 result in a filter time constant of 5 seconds.
MP07_IrTempCtrlGain	LREAL	P parameter of the PID controller • Unit: %/°C / %/°F
MP08_IrTempCtrlResetTime	LREAL	Reset time (I parameter) of the PID controller • Unit: s
MP09_IrTempCtrlRateTime	LREAL	Rate time (D parameter) of the PID controller • Unit: s
MP10_IrTempCtrlIntegral Value	LREAL	Process value of the integral-action component (I) of the controller
MP11_IrTempCtrlDerivative Value	LREAL	Process value of the differential component (D) of the controller
MP12_IrTempCtrlOutPct	LREAL	Correcting variable of the controller • Unit: %
MP13_IrIdentS1ElapsedTime MP14_IrIdentS1TempActGradient MP15_IrIdentS2TempStart MP16_IrIdentS2ElapsedTime MP17_IrIdentS2TempActGradient MP18_IrIdentS3TauDeadAct MP19_IrIdentS3TauDeadMax		see description in section ► Procedure after activating the parameter identification (□ 32)
OP01_IrActTemp	LREAL	Actual temperature value • Unit: °C or °F

6.2

Structure of the access points

L_TP2P_scAP_TemperatureControlBase

The access points (AP) can be used to influence signals. In the initial state, the access points do not have any effect.

Each access point acts as an alternative branch and is activated via an OR operation or a switch.

Designator	Data type	Description
AP01_xActSensorIn	BOOL	Selection of the sensor input signal for the temperature detection • Initial value: FALSE
		TRUE The value of the AP01_lrActSensorIn access point is used for the temperature detection.
		FALSE The sensor input signal of the technology module (nActSensorIn input) is used for the temperature detection.
AP01_lrActSensorIn	LREAL	Alternative value for the sensor input signal, activation with AP01_xActSensorIn = TRUE • Unit: °C or °F • Initial value: 0
AP02_xFilteredTempSet	BOOL	Selection of the temperature setpoint signal for the control • Initial value: FALSE
		TRUE The value of the AP02_lrFilteredTempSet access point is used for the control.
		FALSE The original temperature setpoint from the technology module is used for the control.
AP02_lrFilteredTempSet	LREAL	Alternative value for the temperature setpoint signal, activation with AP02_xFilteredTempSet = TRUE • Unit: °C or °F • Initial value: 0
AP03_xFilteredTempAct	BOOL	Selection of the actual temperature value signal for the control • Initial value: FALSE
		TRUE The value of the AP03_lrFilteredTempAct access point is used for the control.
		FALSE The original actual temperature value from the technology module is used for the control.
AP03_lrFilteredTempAct	LREAL	Alternative value for the actual temperature value signal, activation with AP03_xFilteredTempAct = TRUE • Unit: °C or °F • Initial value: 0
AP04_xControlValuePct	BOOL	Selection of the correcting variable signal for the control • Initial value: FALSE
		TRUE The value of the AP04_lrControlValuePct access point is used for the control.
		FALSE The original correcting variable value from the technology module is used for the control.
AP04_lrControlValuePct	LREAL	Alternative value for the correcting variable signal, activation with AP04_xControlValuePct = TRUE • Unit: % • Initial value: 0

Designator	Data type	Description
AP05_xTempSetFilterTime	BOOL	Selection of the PT1 filter time constant for the temperature setpoint Initial value: FALSE
		TRUE The value of the AP05_lrTempSetFilterTime access point is used for the control.
		FALSE The PT1 filter time constant from the technology module (lrTempSetFilterTime from L_TP2P_scPD_TemperatureControlBase (19)) is used for the control.
AP05_lrTempSetFilterTime	LREAL	Alternative value for the PT1 filter time constant, activation with AP05_xTempSetFilterTime = TRUE - Unit: s - Initial value: 0
AP06_xTempActFilterTime	BOOL	Selection of the PT1 filter time constant for the actual temperature value Initial value: FALSE
		TRUE The value of the AP06_lrTempActFilterTime access point is used for the control.
		FALSE The PT1 filter time constant from the technology module (lrTempActFilterTime from L_TP2P_scPD_TemperatureControlBase (19)) is used for the control.
AP06_lrTempActFilterTime	LREAL	Alternative value for the PT1 filter time constant, activation with AP06_xTempActFilterTime = TRUE - Unit: s - Initial value: 0
AP07_xTempCtrlGain	BOOL	Selection of the P parameter of the PID controller Initial value: FALSE
		TRUE The value of the AP07_lrTempCtrlGain access point is used for the control.
		FALSE The P parameter of the PID controller from the technology module (lrTempCtrlGain from L_TP2P_scPD_TemperatureControlBase (19)) is used for the control.
AP07_lrTempCtrlGain	LREAL	Alternative value for the P parameter of the PID controller, activation with AP07_xTempCtrlGain = TRUE - Unit: %/°C / %/°F - Initial value: 0
AP08_xTempCtrlResetTime	BOOL	Selection of the reset time (I parameter) of the PID controller Initial value: FALSE
		TRUE The value of the AP08_lrTempCtrlResetTime access point is used for the control.
		FALSE The reset time (I parameter) of the PID controller from the technology module (lrTempCtrlResetTime from L_TP2P_scPD_TemperatureControlBase (19)) is used for the control.
AP08_lrTempCtrlResetTime	LREAL	Alternative value for the reset time (I parameter) of the PID controller, activation with AP08_xTempCtrlResetTime = TRUE - Unit: s - Initial value: 0
AP09_xTempCtrlRateTime	BOOL	Selection of the rate time (D parameter) of the PID controller Initial value: FALSE
		TRUE The value of the AP09_lrTempCtrlRateTime access point is used for the control.
		FALSE The rate time (D parameter) of the PID controller from the technology module (lrTempCtrlRateTime from L_TP2P_scPD_TemperatureControlBase (19)) is used for the control.
AP09_lrTempCtrlRateTime	LREAL	Alternative value for the rate time (D parameter) of the PID controller, activation with AP09_xTempCtrlRateTime = TRUE - Unit: s - Initial value: 0

6.3 Use of the access points

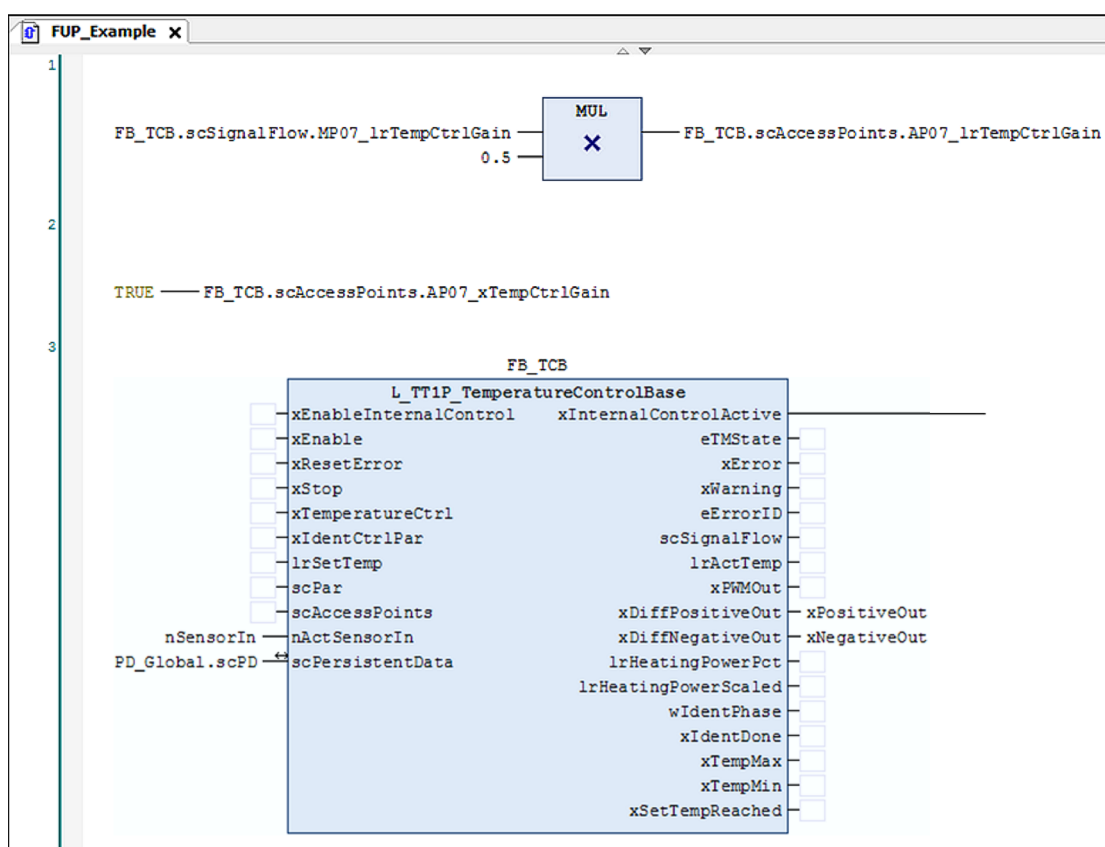
The use of the access points is only required in exceptional cases.

The following application example explains the use of the access points for correcting values of the controller parameters.

Application example

Due to path-specific features, the value of the *lrTempCtrlGain* controller parameter detected by the parameter identification shall not be used directly for the heat section but shall be reduced by the factor '2' (scaling to 50 %).

The required additional external wiring of the **L_TP2P_TemperatureControlBase** function block is shown here:



- In network 1, the controller parameter value is adapted. Here, the *MP07_lrTempCtrlGain* measuring point which provides the value detected by the parameter identification (see [Signal flow diagram: Thermostat \(22\)](#)) is multiplied by the factor '0.5' and is switched to the *AP07_lrTempCtrlGain* access point.
- In network 2, the *AP07_lrTempCtrlGain* is activated by setting the *AP07_xTempCtrlGain* signal to TRUE.
- In network 3, the implementation of the "FB_TCB" module instance is included with minimum circuitry.

7

Basic settings

7.1

Maximum temperature limit

The maximum absolute temperature limit has to be adapted via the *IrTempMaxCriticalAbsolute* parameter (initial value 120 °C) for the heating system and/or the medium to be heated.

The standard values of the other parameters are selected such that the **L_TP2P_TemperatureControlBase** function block can be used without adapting the parameter values.

7.2

Task cycle time

The cycle time of the task in which the **L_TP2P_TemperatureControlBase** function block is used has to be adjusted to the PWM period (*IrPWMPeriod* parameter). Here, the ratio of the times has to be at least 1:100.

Example: In case of a standard PWM period of 1.0 s, the task cycle time must be ≤ 10 ms.

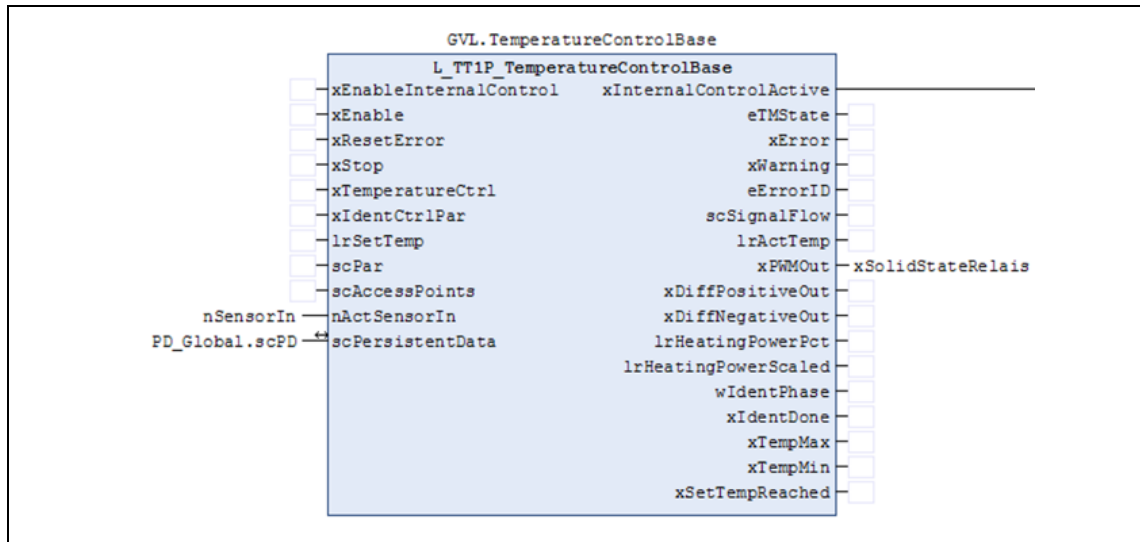
If the ratio of the PWM period to the task cycle time is less than 1:50, the *eErrorID* output displays the warning "17134" (TaskCyclePWMPeriodMismatch).

**"FAST technology modules" reference manual**

Here you can find information on error and warning messages.

7.3 Minimum external wiring for the use of the PWM module

The minimum external wiring of the `L_TP2P_TemperatureControlBase` module for use with the PWM module is shown in the following graphic.



[7-1] Minimum external wiring of the function block

Connection of the thermal sensor

The `nActSensorIn` sensor input can be directly connected to the output signal of an I/O EPM-S405 (I/O system 1000) complete module. As a rule, an output signal can be scaled via the `lrSensorValueGain` parameter so that the temperature can be correctly detected. Since the EPM-S405 delivers an integer temperature value with a resolution of 0.1 °C, 0.1 must also be specified in the `lrSensorValueGain` parameter. This can be monitored at the `lrActTemp` module output.

Connection of a solid state relay for heating control

A heating element is typically controlled via a solid state relay which can be directly connected to the `xPWMOut` module output.

NOTICE

If the heating is to be controlled via a mechanical relay, the PWM period (`lrPWMPeriodparameter`) is to be extended to 10 s, for example.

This can prevent excessive contact actuation and therefore premature closing of the relay.

Persistent data

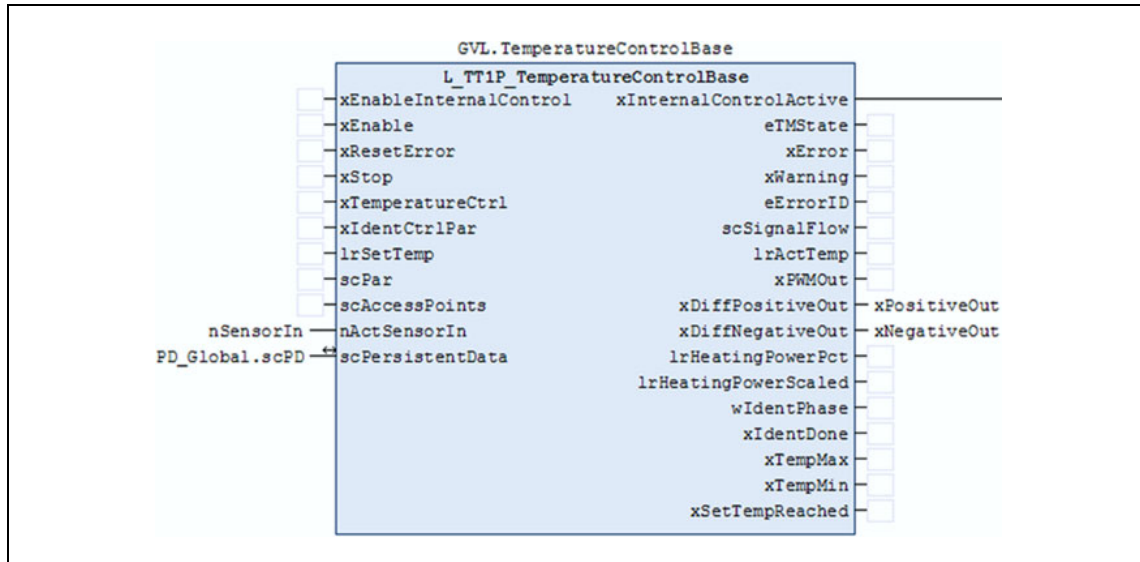
If the controller parameters determined in the course of the parameter identification process are to be maintained even if the controller is restarted, an instance that is part of the persistent variable list has to be connected to `scPersistentData` (VAR_IN_OUT). For the wiring example shown in the figure, this would be as follows:

```
VAR_GLOBAL PERSISTENT RETAIN
    scPD : L_TP2P_scPD_TemperatureControlBase;
END_VAR
```

7.4

Minimum external wiring for the use of the DIFF module

The minimum external wiring of the technology module for the use with the DIFF module is shown in the following illustration.



[7-2] Minimum external wiring of the function block

Connection of the thermal sensor

The *nActSensorIn* sensor input can be directly connected to the output signal of an I/O EPM-S405 (I/O system 1000) complete module. As a rule, an output signal can be scaled via the *lrSensorValueGain* parameter so that the temperature can be correctly detected. Since the EPM-S405 delivers an integer temperature value with a resolution of 0.1 °C, 0.1 must also be specified in the *lrSensorValueGain* parameter. This can be monitored at the *lrActTemp* module output.

persistent data

Generally, the controller parameters detected in the course of the parameter identification process are to be maintained even if the controller is restarted. For this case, an instance that is part of the persistent variable list has to be connected to the *scPersistentData* input/output. For the wiring example shown in figure [7-1] this would be as follows:

```
VAR_GLOBAL PERSISTENT RETAIN
    scPD : L_TP2P_scPD_TemperatureControlBase;
END_VAR
```

8 Description of the technology functions

8.1 Parameter identification

8 Description of the technology functions

8.1 Parameter identification

Before using the **L_TP2P_TemperatureControlBase** function block as thermostat, the parameter identification has to be carried out once. In this process, the system behaviour of the heat section is detected and the optimum values for the controller parameters are derived.

8.1.1 Activation



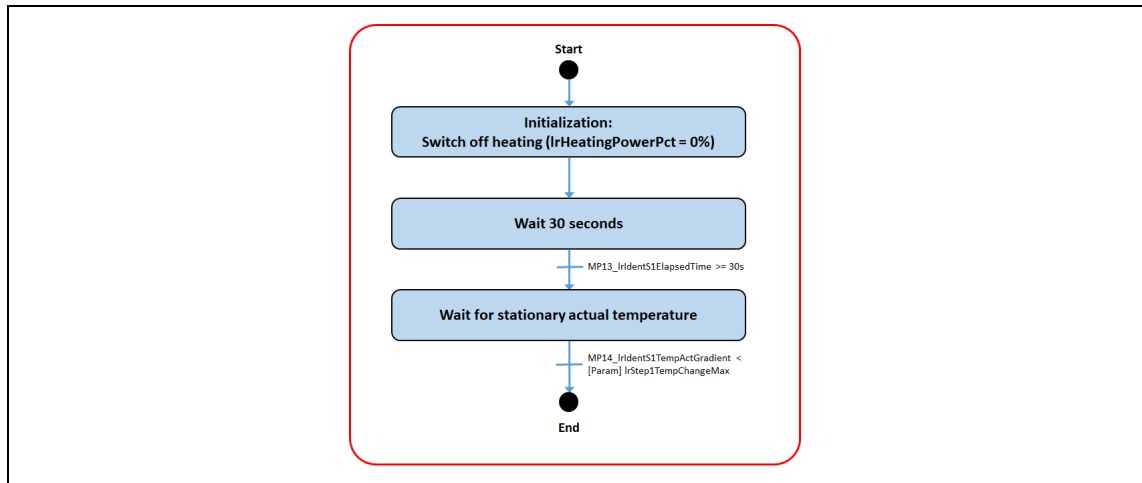
How to activate the parameter identification:

1. Set the *xEnable* input = TRUE.
 - The function block is switched on.
2. Set the *xIdentCtrlPar* input = TRUE.
 - The identification is started.
 - The function block changes to the "@IDENTIFICATION" state (212).

8.1.2 Procedure after activating the parameter identification

Phase 1: Waiting for the ambient temperature

The identification needs to be started at ambient temperature. For this purpose, the change of the current temperature is monitored while the heating element is switched off (correcting variable = 0%). Only when the change rate falls below a limit value (stationary ambient temperature is reached), phase 1 is completed.



[8-1] Status diagram for step 1

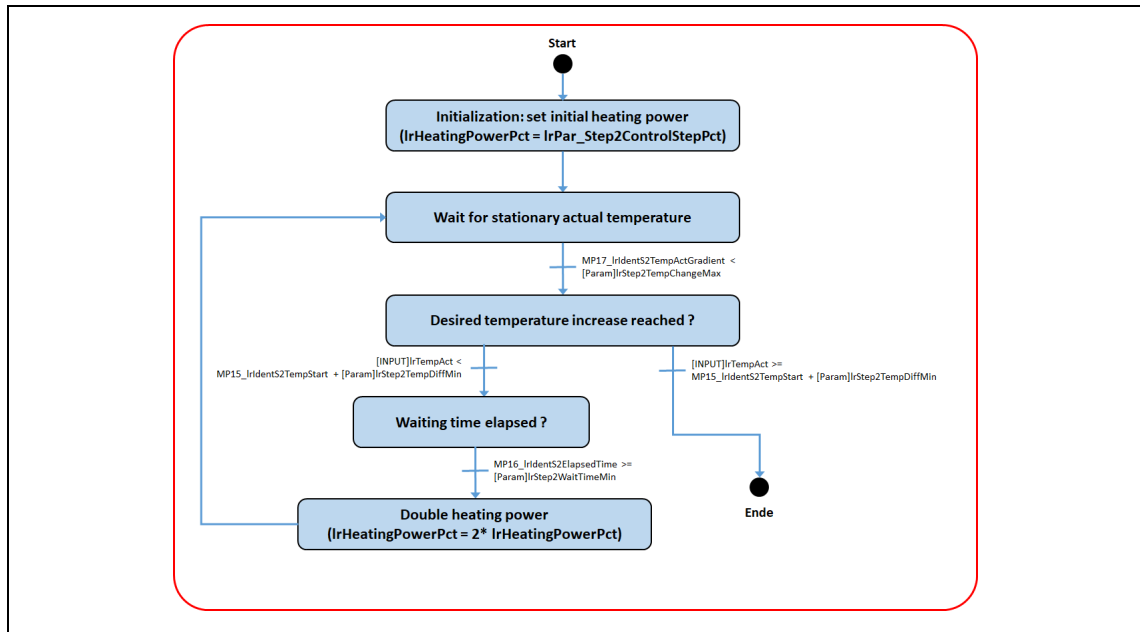
Designator	Data type	Description
MP13_IrIdentS1ElapsedTime	LREAL	Time elapsed since the start of phase 1 of the parameter identification. • Unit: s
MP14_IrIdentS1TempActGradient	LREAL	Current time rate of change of the actual temperature. • Unit: °C/min or °F/min

[8-2] Measuring points in the signal flow

Phase 2: Applying a small correcting variable

The heating element is applied with a small correcting variable and is waited until a stationary temperature has been reached. This action is required to detect the fundamental heating capacity of the heat section and calculate the correcting variable step that is optimal for phase 3.

The current correcting variable for the heating element can be read at the *IrHeatingPowerPct* output (scaled in percent).



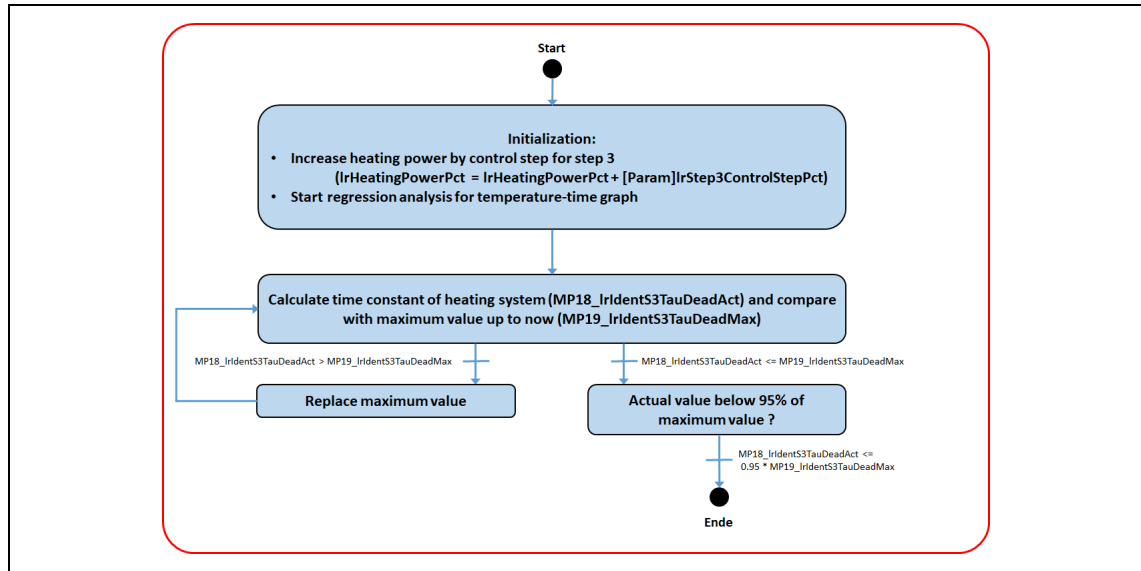
[8-3] Status diagram for step 2

Designator	Data type	Description
MP15_IrIdentS2TempStart	LREAL	Time elapsed since the start of phase 1. • Unit: °C or °F
MP16_IrIdentS2ElapsedTime	LREAL	Time elapsed since the last manipulated variable jump in phase 2. • Unit: s
MP17_IrIdentS2TempActGradient	LREAL	Current time rate of change of the actual temperature. • Unit: °C/min or °F/min

[8-4] Measuring points in the signal flow

Phase 3: Determining the behaviour of the heat section

The heating element is connected with the optimum control value step calculated in phase 2. The relevant characteristic values of the heating section are determined by a regression analysis of the equivalent dead time constant of the heating section.



[8-5] Signal flow for step 3

Designator	Data type	Description
MP18_IridentS3TauDeadAct	LREAL	Current value of the substitute dead time constant of the heating section. • Unit: s
MP19_IridentS3TauDeadMax	LREAL	Maximum value of the substitute dead time constant of the heating section. • Unit: s

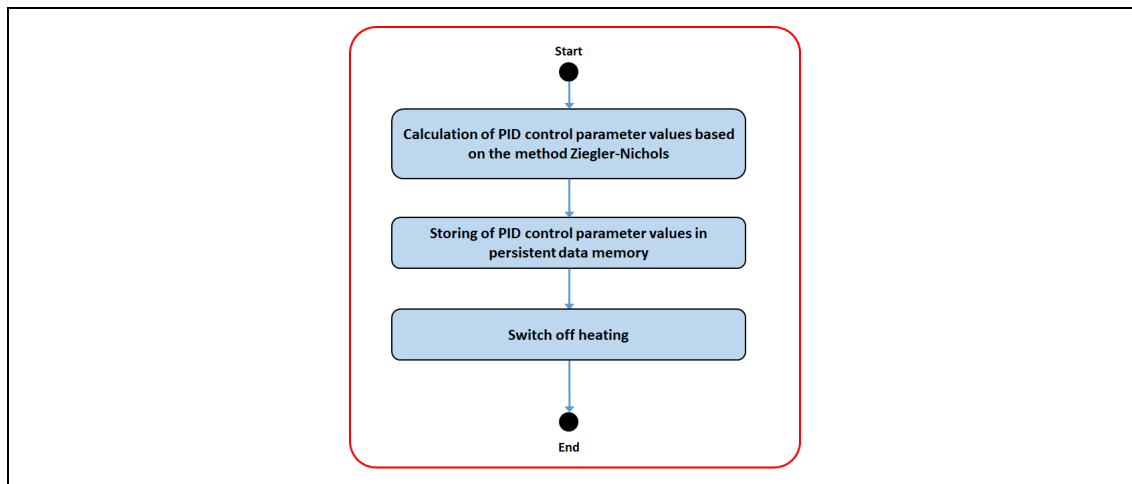
[8-6] Measuring points in the signal flow

Phase 4: Calculating the values for the controller parameters and terminating the identification

The heating element is switched off (correcting variable = 0 %). The characteristic values detected in phase 3 serve to calculate the optimum values for the controller parameters which are saved in the persistent data.

► [Persistent data](#) (📖 37).

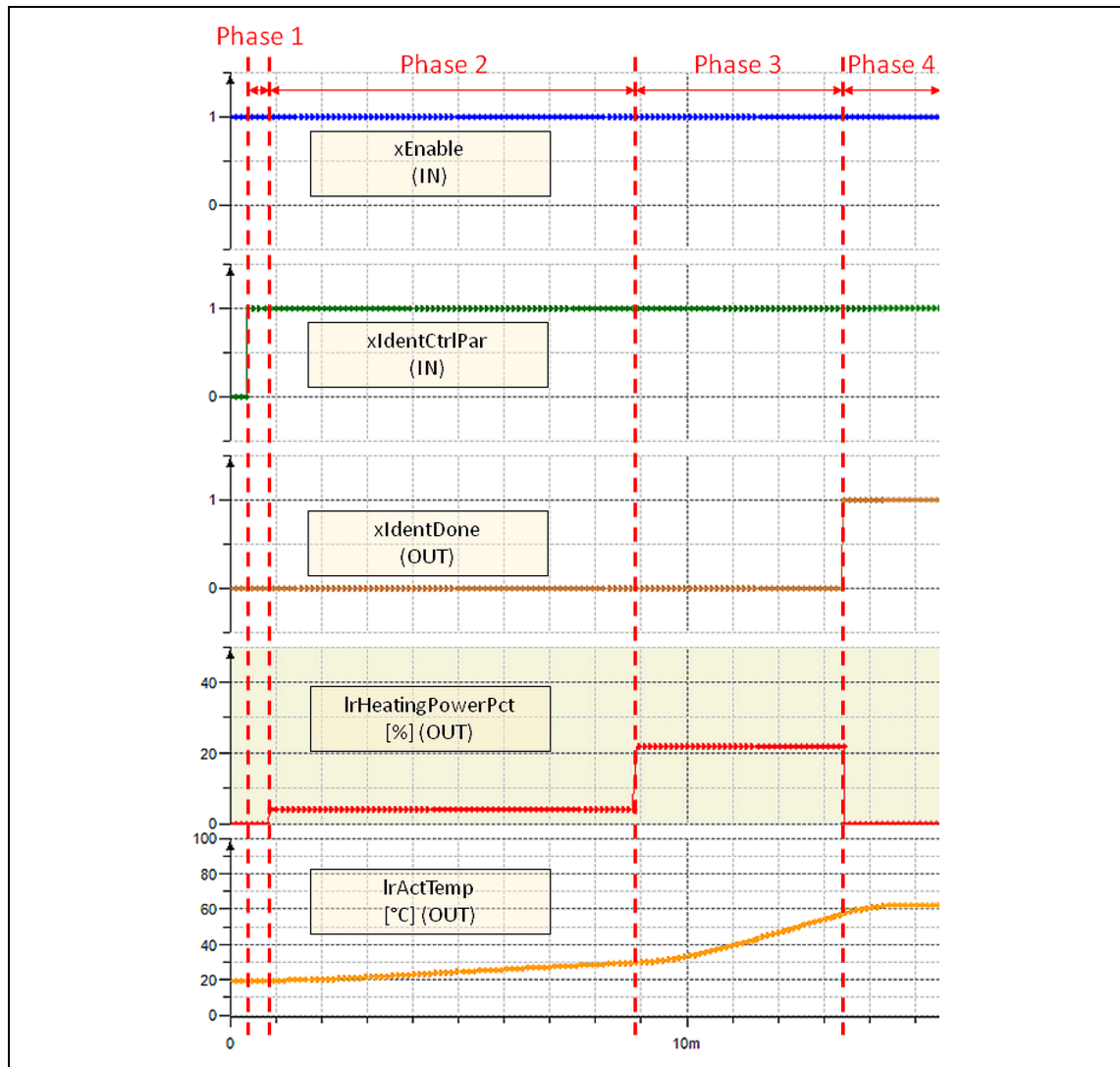
The termination of the parameter identification is signalled: *xlidentDone* output = TRUE



[8-7] Signal flow for step 4

8.1.3 Signal characteristic

The diagram shows the signal characteristic during the previously described [Parameter identification](#) (31).



[8-8] Signal characteristic during parameter identification

8.1.4 Persistent data

The persistent data in the [L_TP2P_scPD_TemperatureControlBase](#) (19) include the following elements:

```
TYPE _scPD_TM_TemperatureControlBase EXTENDS _scPD_TM_Base :  
STRUCT  
    lrTempSetFilterTime : LREAL := 30.0;  
    lrTempActFilterTime : LREAL := 1.0;  
    lrTempCtrlGain : LREAL := 0.0;  
    lrTempCtrlResetTime : LREAL := 100.0;  
    lrTempCtrlRateTime : LREAL := 100.0;  
    dwIdentVersionInfo : DWORD := 0;  
END_STRUCT  
END_TYPE
```

The *dwIdentVersionInfo* parameter has a special meaning. The initial value '0' indicates that so far no [Parameter identification](#) (31) has been carried out. With each parameter identification that has been completed, the values identified for the controller parameters are entered into the LREAL data elements. Afterwards the value of *dwIdentVersionInfo* is increased by '1'.

8.2 Temperature control

If valid values for the controller parameters are available ([Persistent data](#) (📖 37)), the temperature control function can be activated.

8.2.1 Activation

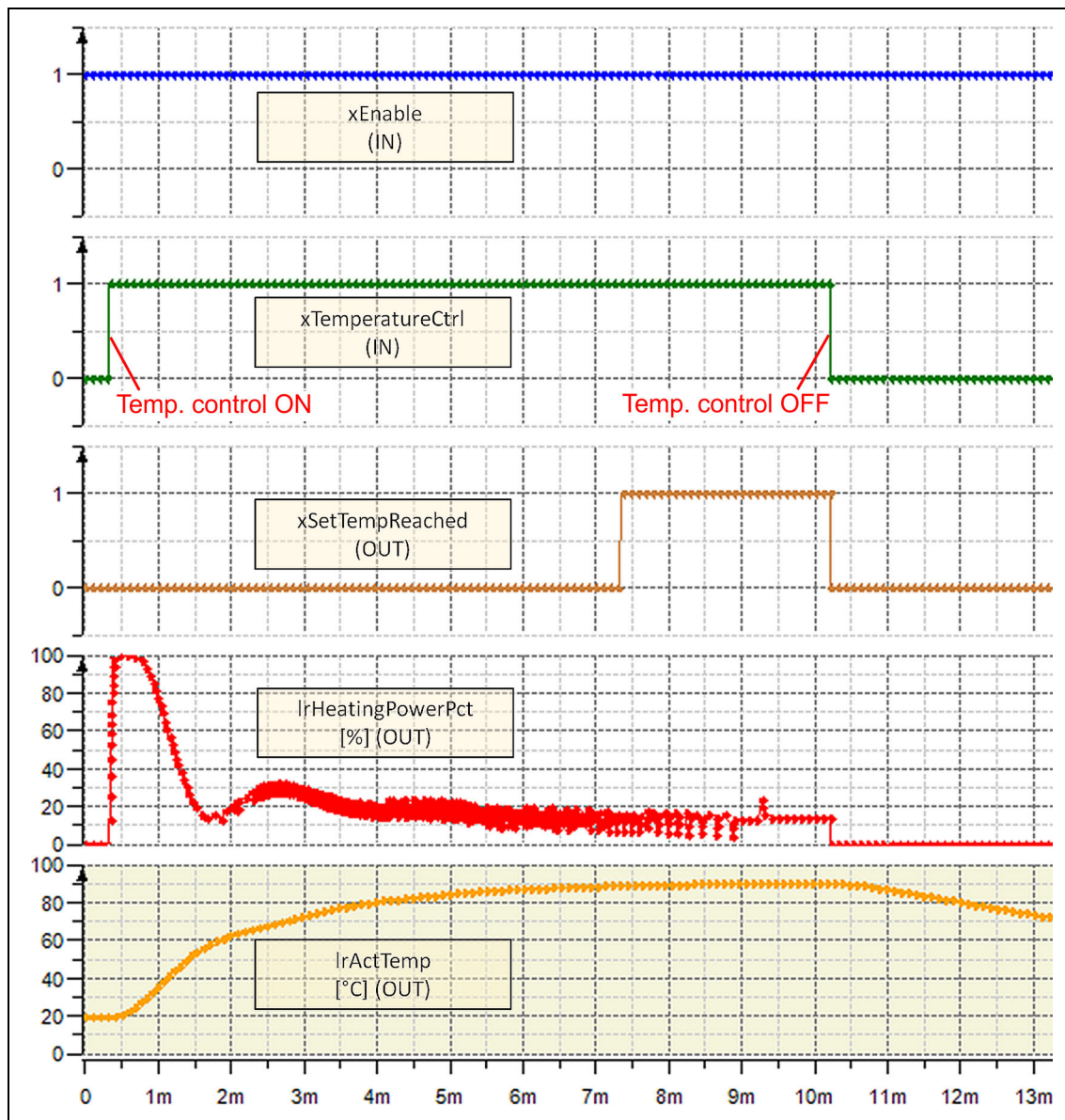


How to activate the temperature control:

1. Set the *xEnable* input = TRUE.
 - The function block is switched on.
2. Set the *eModeTempCtrl* parameter = REGULATION.
3. Define the temperature setpoint to be reached via the *IrSetTemp* input.
4. Set the *xTemperatureCtrl* input = TRUE.
 - The temperature control is started.
 - The function block changes to the "REGULATION" state (211).

8.2.2 Signal characteristic

The sample diagram shows the signal characteristic of the temperature control with a setpoint temperature of 90 °C.



[8-9] Signal characteristic during the temperature control

8.2.3 Temperature control with parameter switching

The configuration of a standard PID controller is characterised by criteria

- which are designed for good disturbance behaviour. (The P-component of the (*IrTempCtrlGain*) controller is defined as the twofold value of the critical route gain according to Ziegler/Nichols.)

and

- which are designed for good guidance behaviour (prevention of overshooting when changing the setpoint temperature, e.g. after switching on the controller. Typically, the time constant of the setpoint filter (*IrTempSetFilterTime*) is defined as the 6-fold value of the controller route dead time for this purpose.)

When the focus is on good guidance behaviour for the assessment of the control response, this design can (in certain cases) result in

- the period until the temperature setpoint is reached for the first time being too long
- The control response being unsteady due to small residual vibrations

In this case, a controller with parameter switching can be used.

In doing so, the current system deviation is assessed and, if necessary, systematic intervention can be made into the controller construction (parameter switching). Switching is done between the following parameter sets:

- Parameter set A "Optimum guidance behaviour"
- Parameter set B "Optimum disturbance behaviour"

Function

Parameter switching is active when *xActivateParameterSwitching* = TRUE.

- After switching on the controller, parameter set A "Optimum guidance behaviour" must first be activated.
- At the time at which the degree of system deviation first falls below a configurable limit value (e.g. $\pm 1\text{ }^{\circ}\text{C}$, *IrDeviationConstraintA* parameter), it switches to parameter set B "Optimum disturbance behaviour".
- Should the degree of system deviation increase again later on (which is typically triggered by a change to the setpoint temperature), it switches back to parameter set A "Optimum guidance behaviour" when a second limit value (e.g. $\pm 3\text{ }^{\circ}\text{C}$, *IrDeviationConstraintB* parameter) is exceeded.

In doing so, the second limit value is consciously selected with a factor 2 ... 4 times greater than the first limit value to prevent excessively frequent switching of the parameter sets near the limit value (hysteresis).

The assessment factors which specify when the parameters should be switched over are defined as parameters:

- *IrGainFactorDisturbanceControl*
- *IrFilterTimeFactorDisturbanceControl*
- *IrGainFactorReferenceControl*
- *IrFilterTimeFactorReferenceControl*

In most common applications, these parameters can be left in the default value.

8.2.4 Temperature control with differential actuator

An application example is a control valve with a 3-point drive which controls the flow of a medium. Such a valve is controlled via 2 electrical connections ("OPEN"; "CLOSED") which directly drive an actuating motor.

"OPEN" activated (controller output *xDiffPositiveOut* = TRUE): Valve opens/is opened.

"OFF" activated (controller output *xDiffNegativeOut* = TRUE): Valve closes/is closed.

If neither of the connections are controlled (*xDiffPositiveOut* or *xDiffNegativeOut* = FALSE, the valve rests in its current position.

The valve position is determined using the runtime of the valve. The runtime is the time the valve drive needs to move the valve from the CLOSED end position to the OPEN end position. The runtime is typically about 60 ... 180 s for valves in heating systems. The runtime is configured in *IrDiffCtrlActuatingTime*.

Notes on the control process:

- Since the *IrDiffCtrlActuatingTime* parameter is a control valve parameter due to the design, it cannot be determined by automatic control parameter identification. It must be determined at least approximately from the technical data, or it can be determined experimentally.
- In order to reduce the strain on the electromechanical components (actuator, relay, etc.), the *IrDiffCtrlSignalTimeMin* and *IrDiffCtrlBreakTimeMin* parameters can be used to specify individual minimum times for the pulse and break duration for the two signals, *xDiffPositiveOut* and *xDiffNegativeOut*.

8.3 Control with constant correcting variable

The special operating mode "control with constant correcting variable" is intended for maintaining an emergency operation if the thermal sensor or its evaluation unit is defective and thus no information on the current actual temperature is available. In this case, a constant manipulating variable serves to control the heating element.

NOTICE

Temperature monitoring is deactivated

In case of control with constant correcting variable, all temperature monitoring modes are deactivated.

Possible consequences

Damages to the heating system or the medium to be heated

Protective measures

Due to previous process monitoring or know-how, select the correcting variable in such a way that the desired actual temperature is set by approximation.

8.3.1 Activation

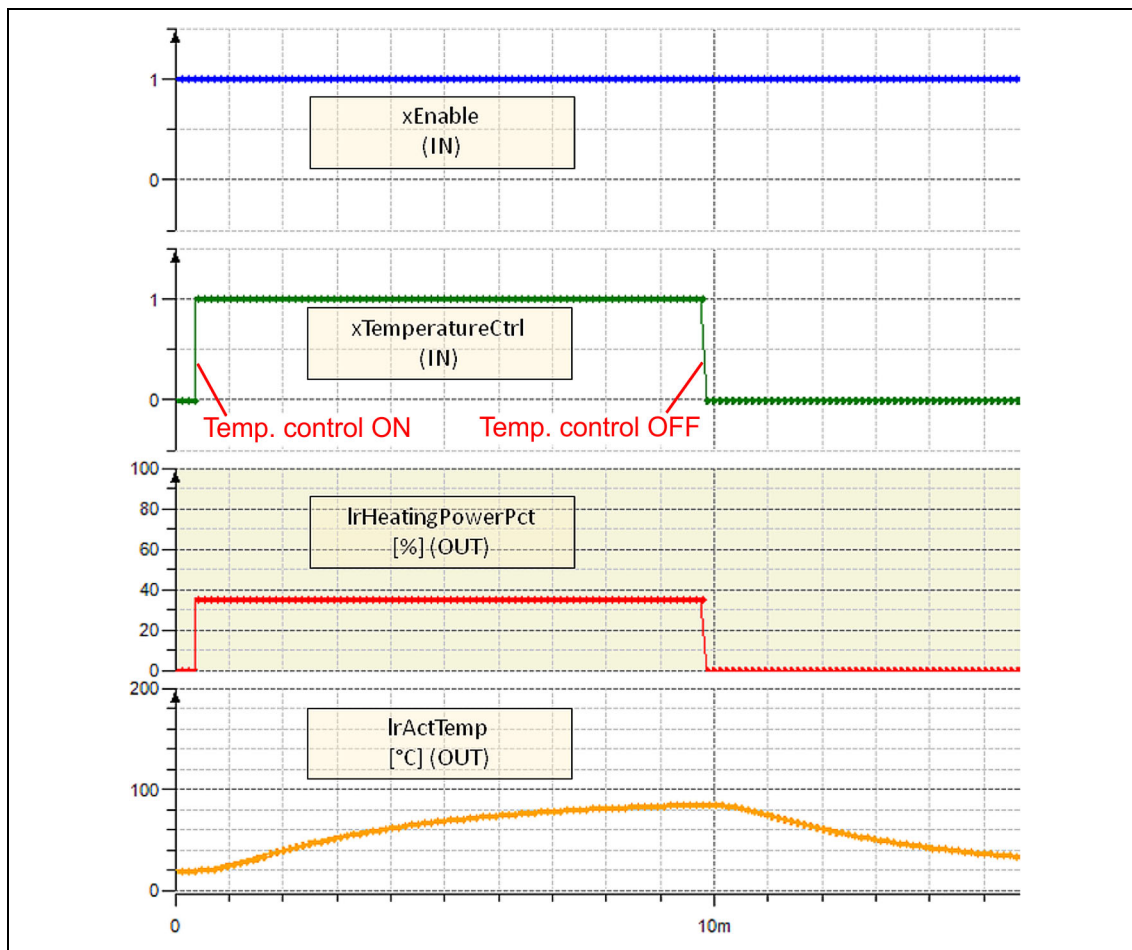


How to activate the control with constant correcting variable:

1. Set the *xEnable* input = TRUE.
 - The function block is switched on.
2. Set *eModeTempCtrl* parameter = CONTROL.
3. Set the value of the correcting variable in the *lrControlValueFixedPct* parameter (initial value is 10 %).
4. Set the *xTemperatureCtrl* input = TRUE.
 - The control is started.
 - The function block changes to the "CONTROL" state (210).

8.3.2 Signal characteristic

The sample diagram shows the signal characteristic of the control with constant correcting variable with a value of 35 %.



[8-10] Signal characteristic during control with constant correcting variable

Symbols

"Temperature Control" functional description [10](#)

A

Access points [25](#)
Access points (use) [27](#)
Activation of the control with constant correcting variable [42](#)
Activation of the parameter identification [31](#)
Activation of the temperature control [38](#)
Actual value conditioning (signal flow diagram) [22](#)
Application notes [7](#)

B

Basic settings [28](#)

C

Connecting the mixing valve [29](#), [30](#)
Connecting the thermal sensor [29](#), [30](#)
Connection of the thermal sensor [29](#), [30](#)
Control with constant correcting variable [42](#)
Control with constant correcting variable (activation) [42](#)
Control with constant correcting variable (signal characteristic) [43](#)
Conventions used [6](#)
Correcting variable conditioning (signal flow diagram) [23](#)

D

DIFF module wiring [29](#), [30](#)
Document history [5](#)

E

E-mail to Lenze [46](#)
eTMState [20](#)

F

Feedback to Lenze [46](#)

I

inputs [12](#)
Inputs and outputs [11](#)

L

L_TP2P_scAP_TemperatureControlBase [25](#)
L_TP2P_scPar_TemperatureControlBase [15](#)
L_TP2P_scPar_TemperatureControlBase parameter structure [15](#)
L_TP2P_scPD_TemperatureControlBase [19](#)
L_TP2P_scSF_TemperatureControlBase [24](#)
L_TP2P_TemperatureControlBase [11](#)
L_TP2P_TemperatureControlBase function block [11](#)
Layout of the safety instructions [7](#)

M

Maximum temperature limit [28](#)
Minimum external wiring [30](#)

O

outputs [13](#)
Overview "Temperature Control" [10](#)

P

Parameter identification [31](#)
Parameter identification (activation/procedure) [31](#)
Parameter identification (persistent data) [37](#)
Parameter identification (signal characteristic) [36](#)
Persistent data L_TP2P_scPD_TemperatureControlBase [19](#)
Persistent data of the parameter identification [37](#)
Procedure after activation the parameter identification [32](#)
Procedure of the parameter identification [31](#)

S

Safety instructions [8](#)
Setpoint conditioning (signal flow diagram) [22](#)
Signal characteristic for control with constant correcting variable [43](#)
Signal characteristic for parameter identification [36](#)
Signal characteristic for temperature control [39](#)
Signal flow diagram - correcting variable conditioning [23](#)
Signal flow diagram - setpoint/actual value conditioning [22](#)
Signal flow diagram of thermostat [22](#)
State Machine [20](#)
State machine [20](#)
States [20](#)
States of the eTMState output [20](#)
Structure of the access points
L_TP2P_scAP_TemperatureControlBase [25](#)
Structure of the signal flow
L_TP2P_scSF_TemperatureControlBase [24](#)

T

Target group [4](#)
Task cycle time [28](#)
Technology module functions (overview) [10](#)
Temperature control [38](#)
Temperature control (activation) [38](#)
Temperature control (signal characteristic) [39](#)
Thermostat (signal flow diagram) [22](#)

U

Use of the access points [27](#)

V

Variable names [6](#)

W

Wiring for the use of the DIFF module [29](#), [30](#)
Wiring of the technology module [29](#), [30](#)

Z

Zustände [20](#)



Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

