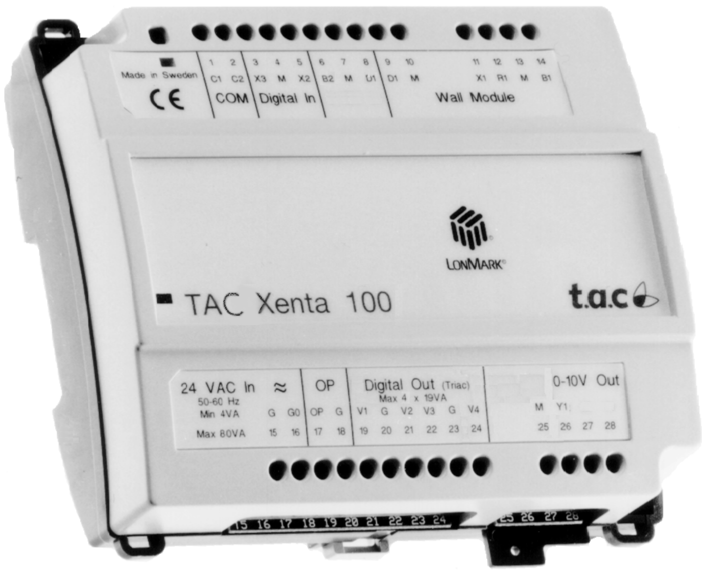
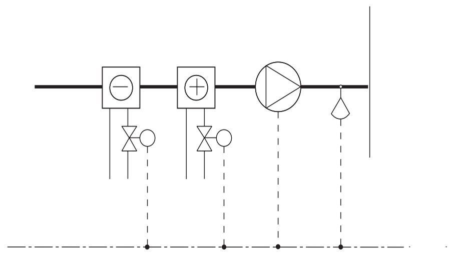
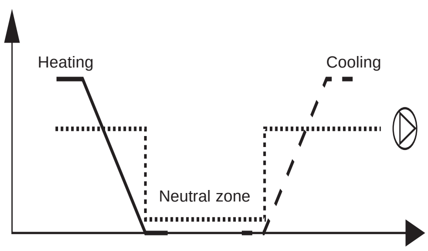
Zone Controller
TAC Xenta 101

Handbook

TAC improves indoor climate
and reduces operating costs.







0-004-7513-1 (GB), 1999-08-16

Foreword

This is the technical handbook for the Xenta 101 controller, a fan coil controller for offices and other larger buildings.

In this second edition of the handbook, sections that were earlier complicated to the user, have been made clearer, and most of the content has been reorganized. The trouble-shooting section has been made into its own chapter, and there are now appendices in the end of the handbook; one containing setpoint calculating examples, and one containing a commissioning protocol which can be used together with chapter 3 when commissioning.

The programs in TAC Xenta 101 now have new versions. For both the system program and the application program in the controller, the versions are 1.10. If there is a service replacement in the system, all variable bindings–if the controller is run on a network–must be remade when an older or newer version of the controller is fitted. This is because the controller has got a new “Standard Program ID”. There are also three new network variables.

Revisions list

Part number	*Comment*	*Editor*	*Date*
0-004-7514-0	First edition.	KRRO	1997-09-11
0-004-7514-1	Second edition. New versions (1.10) of system- och application programs.	SUWA	1999-08-16

Zone controller TAC Xenta 101

Handbook

Reservations for changes.

Contents

1 Introduction

1.1 The content of the handbook

- **Chapter 1 Introduction,**
  gives an overview over the structure of this handbook, additional information about the product, and has a short terminology section.
- **Chapter 2 The zone controller TAC Xenta 101**,
  describes, among other things, the wall module, and briefly the controller's functions. It shows applications for the four different types of TAC Xenta 101.
- **Chapter 3 Installation**,
  contains instructions on mechanical and electrical installation of the controller, and instructions on commissioning and network installation.
- **Chapter 4 Configuration parameters**,
  describes the setting of the zone controller's configuration parameters.
- **Chapter 5 Functional description**,
  gives detailed information about the zone controller's basic functions, operating modes, and other functions.
- **Chapter 6 Trouble-shooting during operation and commissioning,**
  contains trouble-shooting measures you can use to find and fix possible faults in the system.
- **Chapter 7 Technical data**,
  lists all technical data and dimensions for TAC Xenta 101.
- **Chapter 8 Communication**,
  describes the zone controller's communication with other units via the network by means of network variables.
- **Appendix A**, **Setpoint calculation**
  contains calculating examples for the setpoint calculation in chapter 5.
- **Appendix B**, **Commissioning protocol**
  contains a commissioning protocol, which can be used together with chapter 3 during installation and commissioning.

- **Index and Reply form**,
  are in the end of the handbook. Use the index to make your search for information easier, and the reply form to let us know whether there is something wrong or unclear in this handbook.

1.2 Documentation

Enclosed documentation

TAC Xenta 101 is delivered with an installation instruction for each of the following types of controllers:

- *TAC Xenta 101-1VF*,
  Installation instruction, part number 0FL-3853
- *TAC Xenta 101-1VFC,*
  Installation instruction, part number 0FL-3854
- *TAC Xenta 101-2VF,*
  Installation instruction, part number 0FL-3855
- *TAC Xenta 101-2VFC,*
  Installation instruction, part number 0FL-3856

Other documentation

There is additional information about TAC Xenta 101 in the following documents:

- *Data sheet for TAC Xenta 101-1VF,*
  part number 0-003-1743
- *Data sheet for TAC Xenta 101-1VFC,*
  part number 0-003-1388
- *Data sheet for TAC Xenta 101-2VF,*
  part number 0-003-1605
- *Data sheet for TAC Xenta 101-2VFC,*
  part number 0-003-1608
- *Data sheet for ZS 101–ZS 105,*
  part number 0-003-1661. Here the wall modules are described.
- *TAC Xenta Network Guide,*
  part number 0-004-7460. Here you can find additional information on network installation.
- *TAC Xenta OP Handbook,*
  part number 0-004-7506. Here you find information on how to use TAC Xenta OP together with TAC Xenta 101 and the wall modules.
- *TAC Xenta, Zone System Guidelines*
  part number 0-004-7637. Here you find information on how the zone system is built with TAC Xenta-components.

- *TAC Xenta 101 Handbook,*
  part number 0-004-7513-0. Here you find information on the earlier version of the zone controller.

All the above mentioned documents can be found on the internet at www.tac.se or can be ordered from your nearest TAC service point.

1.3 Terminology

In this handbook there are some abbreviations and terms which are specific for the zone controller's applications and network communication. Therefore, the most common terms have been gathered, together with a short explanation, in the list below.

neuron .... communication processor with built-in protocol

node .... communication unit on the network

SNVT .... Standard Network Variable Type

nvixxx .... variable which gets its value from another unit on the network

nvoxxx .... variable which value is sent out to another unit on the network

ncixxx .... configuration parameter; variable which gets its value from another unit on the network and which keeps it during a power failure

service pin .... function which can be used during installation on the network

wink .... confirmation that the connection to a controller via the network is working (a light emmitting diode is lit for appr. 15 seconds)

LNS .... LonWork Network Services. System tool for installation, configuration and maintenance of LonWorks network

Blank page

2 The zone controller TAC Xenta 101

2.1 General

The zone controller TAC Xenta 101 is intended for fan-coil applications in offices and other large buildings. The fan-coil controller keeps a constant temperature in a given zone by controlling the temperature and air flow of circulating air through a fan-coil unit. In the TAC Xenta 101 series there are four types of controllers: 1VF, 1VFC, 2VF, 2VFC.

Properties	TAC Xenta 101			
	1VF	1VFC	2VF	2VFC
One valve fan coil system, combined heating/cooling coil	X	X		
Two valve fan coil system, separate heating/cooling coils			X	X
Fan on/off control	X		X	
Fan speed control		X		X

The controller's basic functions

All controller types have a number of built-in functions which handle the normal control situation. There are four operating modes to choose from (comfort, economy, bypass, and off) and five modes to force the controller (only heating allowed, only cooling allowed, only fan allowed, auto, and off). Measuring the zone temperature is made by means of a permanent thermistor sensor or a temperature node connected to the network. Setpoint calculation is made according to special methods. Depending on the type of controller, the fan control is either an on/off or a speed control. For the controllers with one valve, a water temperature sensor can be connected directly to the controller.

There is a detailed functional description of all the basic functions in chapter 5.2.

More about functions

Apart from the controller's basic functions, there are additional possibilities to control the climate in the zone. In section 5.3, these are described in detail, and also which external functions that may be connected, e.g. window contact sensor and occupancy sensor.

Communication possibilities

The controller can work either as a stand-alone unit, without being connected to a network during operation, or be a part of a larger system with several other units such as TAC Xenta 300/400 and other zone controllers in the TAC Xenta family (figure 2.1). A detailed description of how units work together in a larger zone system, is found in “Zone Systems Guidelines”, part number 0-004-7637.

TAC Vista is an excellent tool for reading variables and a configuration tool during commissioning and/or operation. When TAC Vista is not a part of the system, reading and configuration of variables can be made from the operating panel TAC Xenta OP, version 3.11 or later.

The controller is LonMark® approved and communicates on a LonTalk® TP/FT-10 network via a twisted-pair, unpolarized cable. If you want to know more about the LonWorks® technology, the internet addresses www.echelon.com or www.lonmark.org are good sources of information.

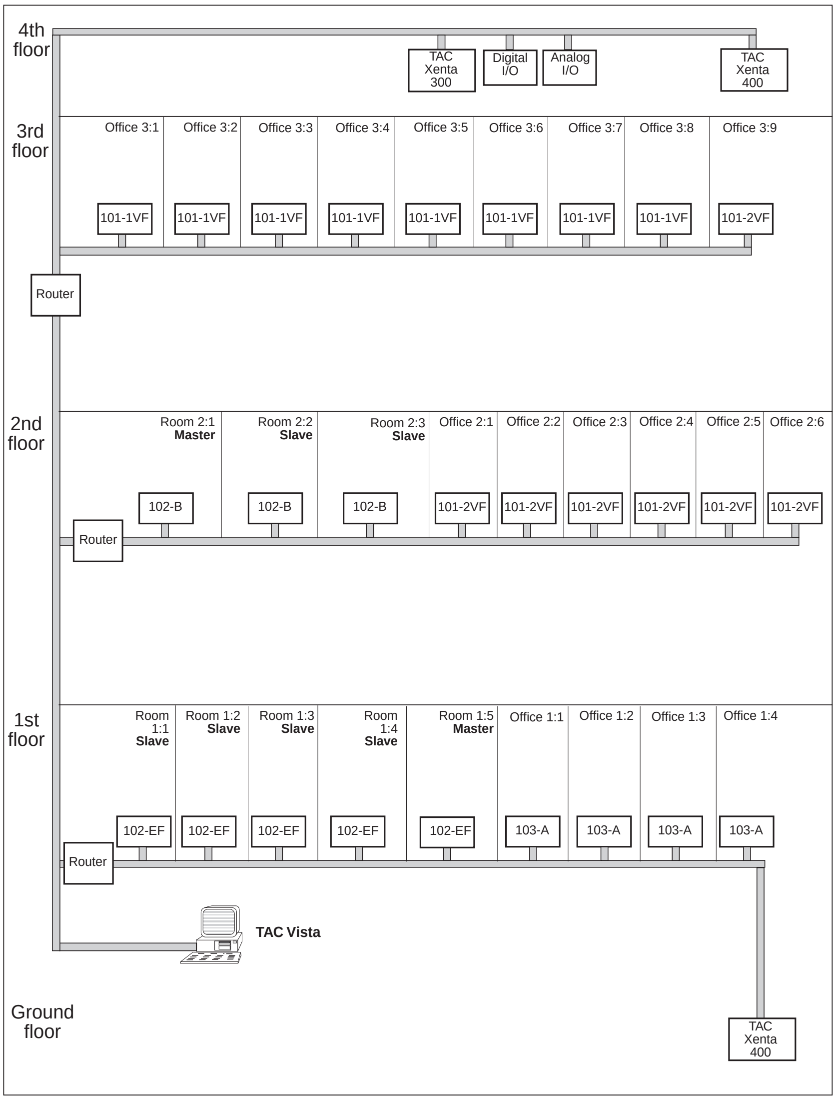


Figure 2.1 Zone controller in a larger system together with TAC Vista

2.2 Wall modules

In the controlled zone, there is usually a wall module from the ZS 100 series, which measures the temperature. The wall modules ZS 101–ZS 105 may very well be used together with all four controller types. On the wall module (figure 2.2) there are e.g. a setpoint knob and a bypass key with setting possibilities.

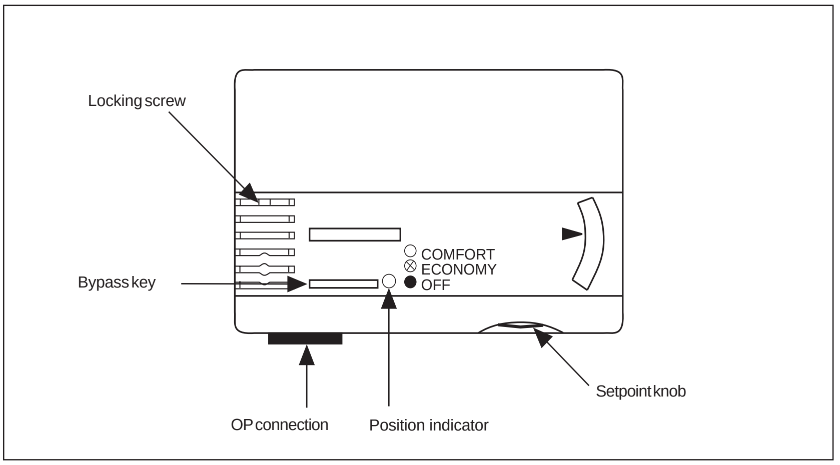


Figure 2.2 Zone sensor in the ZS100 series

The setpoint knob is used to adjust the zone temperature setpoint with a maximum of ± 5°C.

The bypass key is used to change the operating mode, and by pressing the key, an internal timer in the controller, which runs for two hours, is started. Read more about different operating modes and ways to force the controller in sections 5.2.1–5.2.2.

On all ZS 100 wall modules, the current operating mode is indicated by the position indicator (red light emitting diode) as follows:

- Steady light: Comfort or bypass mode
- Slow blinking: Economy mode
- Fast blinking for 15 s: Answer for “wink” command. Confirmation that the OP is connected to the correct controller
- Off: Other operating modes

There is additional information on the wall modules and how the temperatures can be adjusted locally in the zone by means of the keys in “Data sheet for ZS 101–ZS 105”, part number 0-003-1661.

2.3 Applications

2.3.1 General

To all Xenta 101 controllers,

• a window contact sensor to stop the heating and cooling functions, should a window be opened, can be connected. An occupancy sensor can detect the presence of a person in the controlled zone and change the controller from economy to comfort mode.

• the fan is off in economy and off mode, if there is not a need for heating or cooling. In those cases, the fan is running.

• if you connect a water temperature sensor, you can have an automatic change between heating/cooling (only for controller with a combined heating/cooling coil).

• there is also a possibility of an even better control, if you connect a discharge air temperature sensor. When such a sensor is connected, the controller controls the discharge air temperature and the zone temperature in cascade.

The window contact sensor, occupancy sensor, and cascade control of the discharge air and zone temperatures, are described in detail in section 5.3.2–5.3.4.

2.3.2 The zone controller TAC Xenta 101-1VF

This controller (figure 2.3) has one valve output connected to a combined heating/cooling coil. The zone temperature is kept constant because the controller controls the heating/cooling coil. The changeover between heating and cooling is accomplished either by the operating mode being forced via the network, or by using the water temperature sensor.

The fan is turned on or off by means of a relay output. In comfort and bypass mode the fan is running constantly.

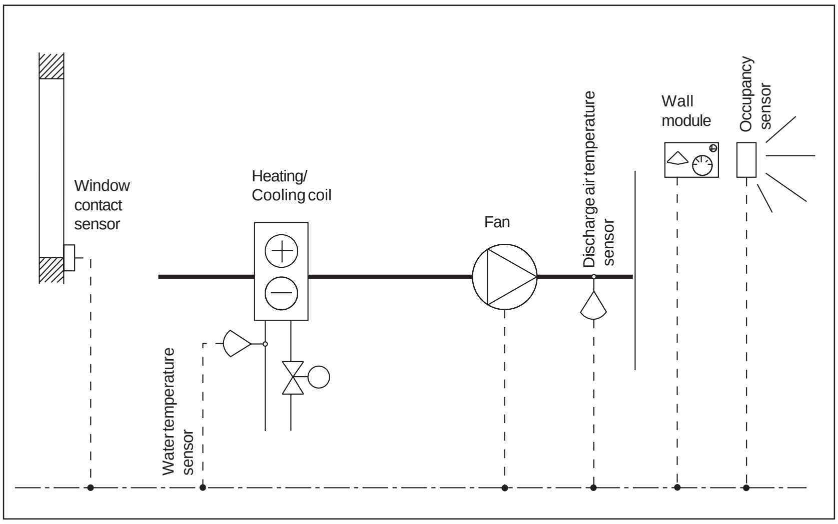


Figure 2.3 Control application for TAC Xenta 101-1VF

2.3.3 The zone controller TAC Xenta 101-1VFC

This type (figure 2.4) works as the 1VF type except from the fan control. Here the fan speed is controlled by a control signal by means of an analogue output.

In the comfort and bypass modes the fan is running on lowest speed and the control is made with the heating/cooling valve. When the valve is completely open, the zone temperature is controlled by the fan speed. In the other modes, the fan is at a stand still.

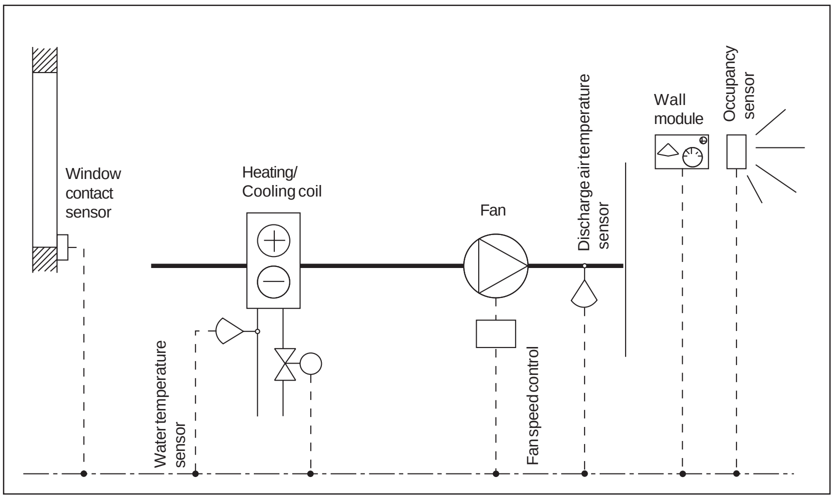


Figure 2.4 Control application for TAC Xenta 101-1VFC

2.3.4 The zone controller TAC Xenta 101-2VF

This controller (figure 2.5) has two valve outputs connected to separate heating and cooling coils. The zone temperature is kept constant by the controller controlling either the heating or the cooling coil.

The fan is turned on or off by means of a relay. In comfort and bypass mode the fan is running constantly.

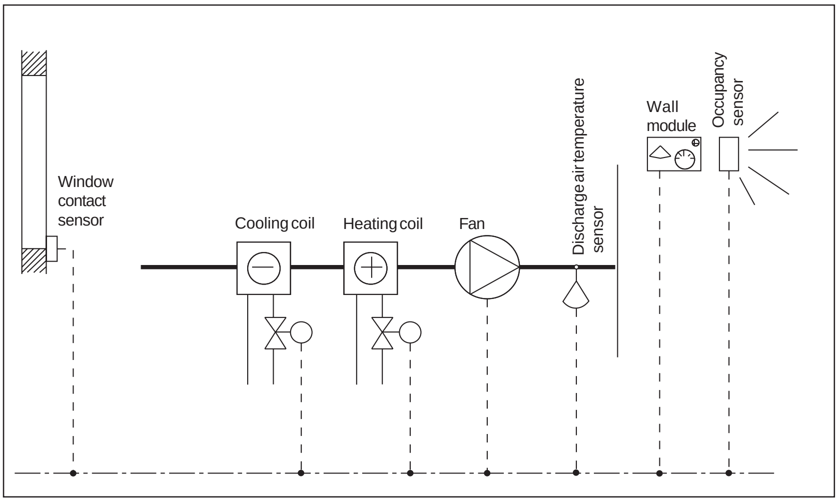


Figure 2.5 Control application for TAC Xenta 101-2VF

2.3.5 The zone controller TAC Xenta 101-2VFC

This controller (figure 2.6) has two valve outputs connected to separate heating and cooling coils. The zone temperature is kept constant by the controller controlling either the heating or the cooling coil.

The controller works as the 2VF type except from the fan control. Here the fan speed is controlled by a control signal by means of an analogue output.

In the comfort and bypass modes the fan is running on lowest speed and the control is made with the heating or cooling valve. When one of the valves is completely open, the zone temperature is controlled by the fan speed.

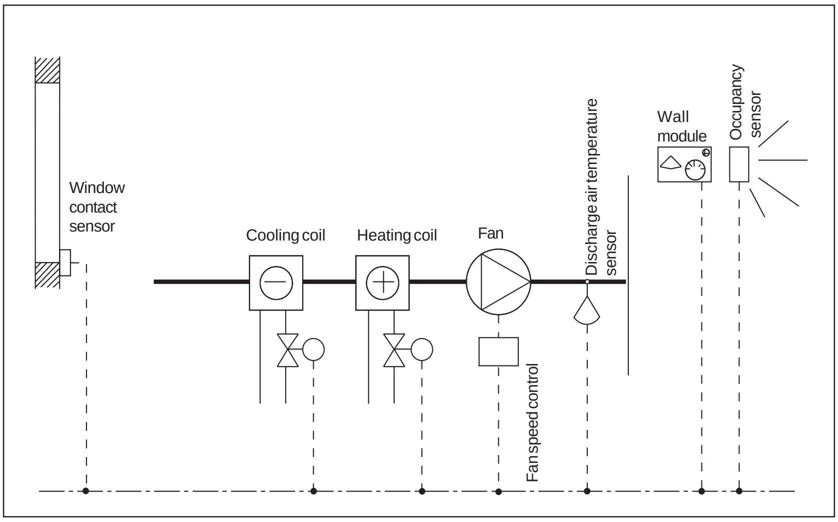


Figure 2.6 Control application for TAC Xenta 101-2VFC

3 Installation

3.1 Mechanical installation

3.1.1 Fitting

TAC Xenta 101 can either be snapped onto a DIN rail (figure 3.1) or fastened with two screws to a level surface (figure 3.2). On controllers which control equipment with a 230 V supply, a semi-protection covering the relay terminals should be fitted (figure 3.3). A semi-protection is delivered with these controllers.

To fasten the controller onto a DIN rail:

1. Place the controller on the top of the rail as is shown by arrow 1.
2. Turn the controller downwards until it snaps onto the rail as is indicated by arrow 2.

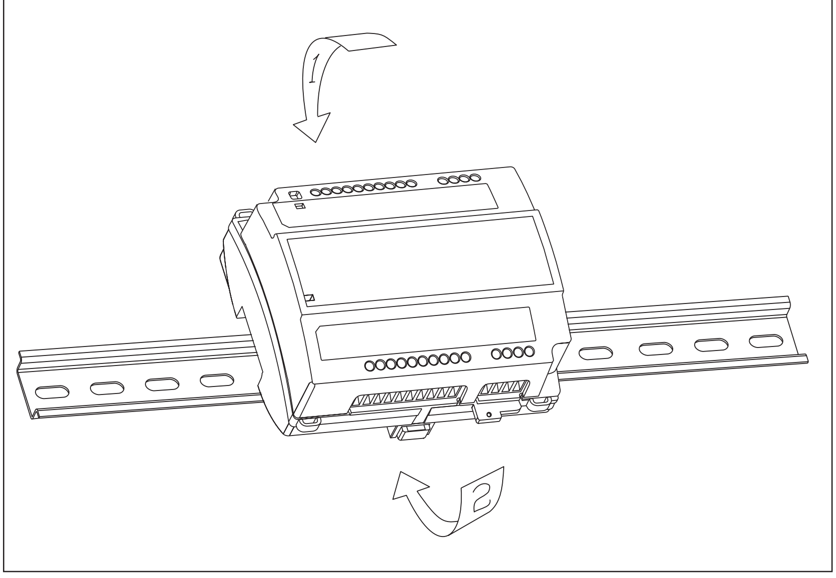


Figure 3.1 TAC Xenta 101 fastened on a DIN rail

3. To remove, place a screwdriver in the lock on the bottom of the controller and pull down. It is then possible to lift the controller diagonally upwards and off the rail.

Fastening the controller on a level surface:

Use the two sockets provided for fastening the controller; the maximum screw size is M4 or ST 3,5. The head of the screw should not exceed 7,5 mm in diameter.

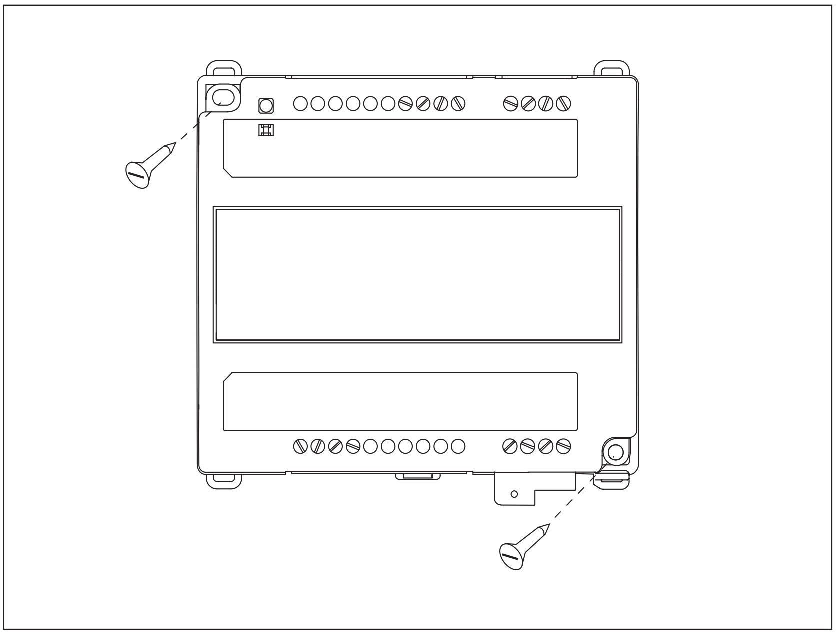


Figure 3.2 TAC Xenta 101 fastened on a level surface

To fit a semi-protection

When the cables have been connected, the protection is fitted with the enclosed screw.

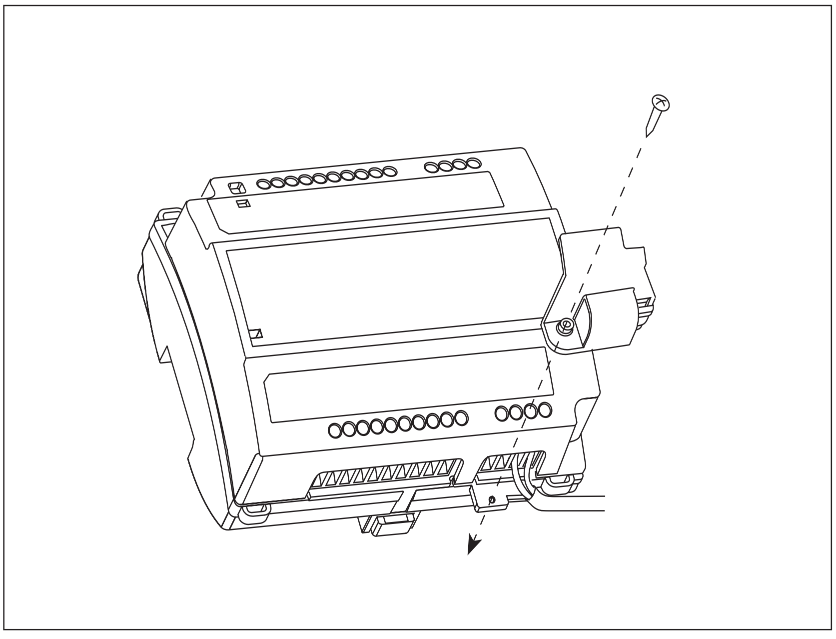


Figure 3.3 Fitting of a semi-protection

3.2 Electrical installation

3.2.1 General

> ***Warning!*** All 230 V supply cables must be installed by authorised electricians.

1. Each controller or group of controllers must be fitted with max. 6 A fuses.
2. Secure the cables to the controller by means of clamps or similar, to limit their mobility.
3. Strap wires or shrink-to-fit tubes must be fitted to make sure that loose 230 V cables cannot get in contact with ELV cables–supply or signal cables–and vice versa.
4. It must be simple to break the power supply for the controller or for the complete installation.
5. Connect U1 and M with a jumper when no water temperature sensor is connected to the terminals.
6. When several Xenta controllers are supplied from a common transformer, it is important that all G’s are connected with each other and that all G0’s are connected with each other. They must not be interchanged. An important exception: G0 on the wall module should not be connected with the other G0’s. Instead it should be connected to the terminal OP on the controller.

   At the transformer, G0 should be connected to protective earth. This is to get an grounding point for interference diversion.
7. Connect the two M terminals to the wall module to get the specified measuring accuracy for the room temperature.

Safety standard

Transformers supplying the controller must comply to the safety standard EN 60 742 or any other relevant safety standard for ELV, 24 V AC. When equipment with a power supply of its own is connected, this power supply must also comply with this norm.

Cable lengths

For information on communication cable lengths, see TAC Xenta Network Guide, part number 0-004-7460. For all other cables, maximum length is 30 m and min. area is 0,7 mm^2 .

The wall modules ZS 101–ZS 104

Primarily, ZS 101–ZS 104 are intended for use with the Xenta 101. The wall module ZS 105 can also be used, but then the fan switch on this unit is not used. The wiring diagram shows how to connect ZS 104, as this is the model which has all connections.

Connection terminals

The designation of the connection terminals can be seen in two places on the controller: on the edge of the printed circuit board, and on the label on the front of the controller.

Termin.	Design.	Function	Type
1	C1	TP/FT-10 communication channel	-
2	C2	TP/FT-10 communication channel	-
3[*5]	X3	Window contact (Closed contact=closed window)	Digital input
4	M	Measurement neutral	-
5[*5]	X2	Occupancy sensor	Digital input
6	B2	Discharge air temperature sensor	Thermistor input
7	M	Measurement neutral	-
8[*4]	U1	Water temperature sensor or jumper	Thermistor input
9	D1	LED on wall module	Digital input
10	M	Measurement neutral	
11	X1	Bypass key on wall module	Digital input
12	R1	Setpoint adjustment on wall module	10 kΩ linear potentiometer
13	M	Measurement neutral	-
14	B1	Room temperature sensor	Thermistor input
15	G	24 V AC (G)	Input
16	G0	24 V AC (G0)	Input
17[*6]	OP	24 V AC supply for TAC Xenta OP	-
18	G	24 V AC supply for TAC Xenta OP	-
19	V1	Heating/cooling actuator: increase Thermo actuator: on/off parallel V2	Triac
20	G	24 V AC (G) supply for V1, V2	-
21	V2	Heating/cooling actuator: decrease Thermo actuator: on/off parallel V1	Triac
22[*1]	V3	Cooling actuator: increase Thermo actuator: on/off parallel V4	Triac
23	G	24 V AC (G) supply for V3, V4	-
24[*1]	V4	Cooling actuator: decrease Thermo actuator: on/off parallel V3	Triac
25	M	Measurement neutral	-
26[*2]	Yl	Fan speed	Analogue output
27[*3]	K1	Fan control, **max 230 V AC**	Relay
28	KC1	Supply for K1	-

[*1] Only controllers 2VF and 2VFC

[*2] Only controllers 1VFC and 2VFC

[*3] Only controllers 1VF and 2VF

[*4] Only controllers 1VF and 1VFC

[*5] See chapter 4 Configuration parameters

[*6] Connected to G0 on the wall module. Must *not* be connected to G0 on the controller.

3.2.2 Wiring of TAC Xenta 101-1VF

Note! Read section 3.2.1 “General” before you connect the cables according to the wiring diagram in figure 3.4.

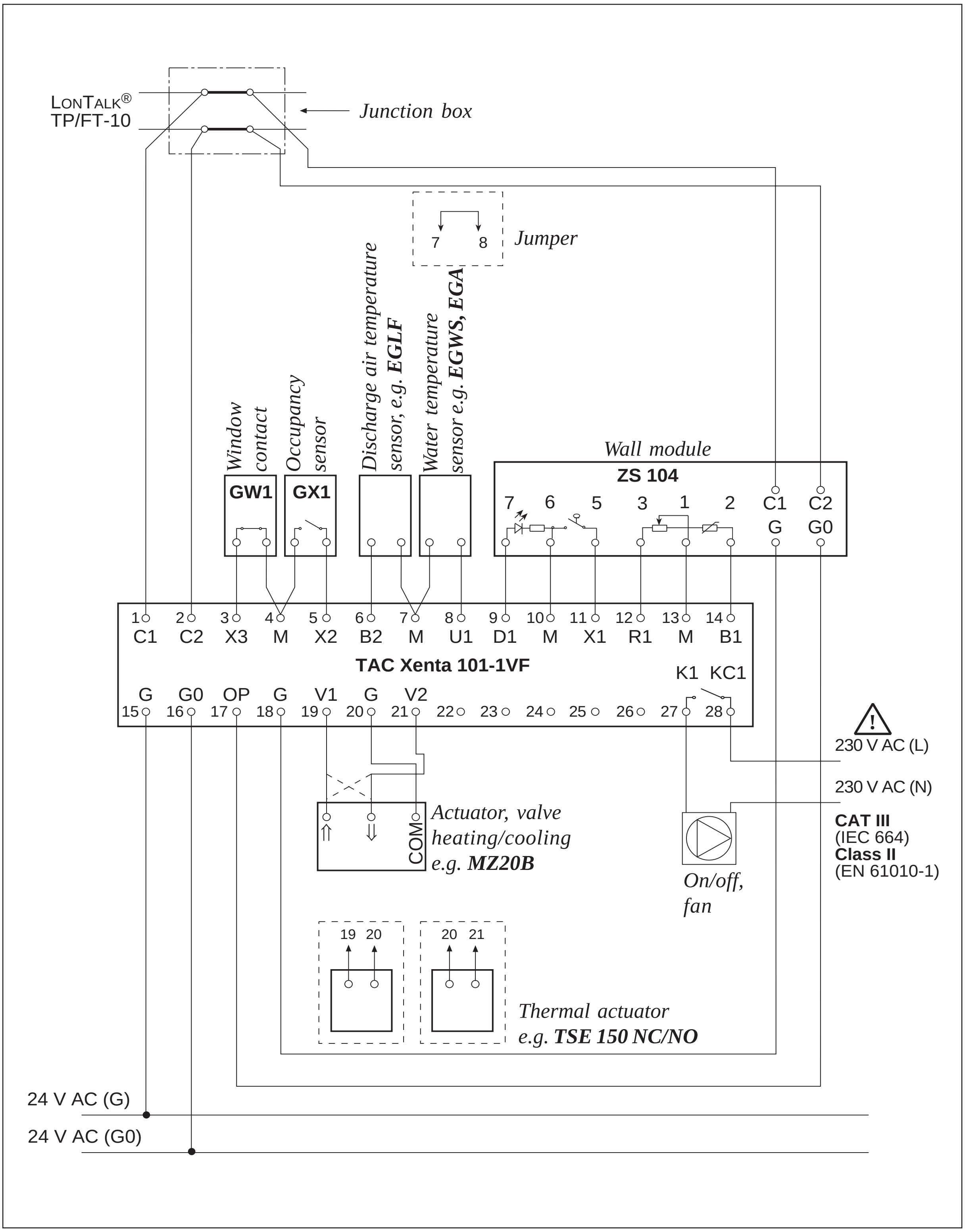


Figure 3.4 Wiring of TAC Xenta 101-1VF

3.2.3 Wiring of TAC Xenta 101-1VFC

Note! Read section 3.2.1 “General” before you connect the cables according to the wiring diagram in figure 3.5.

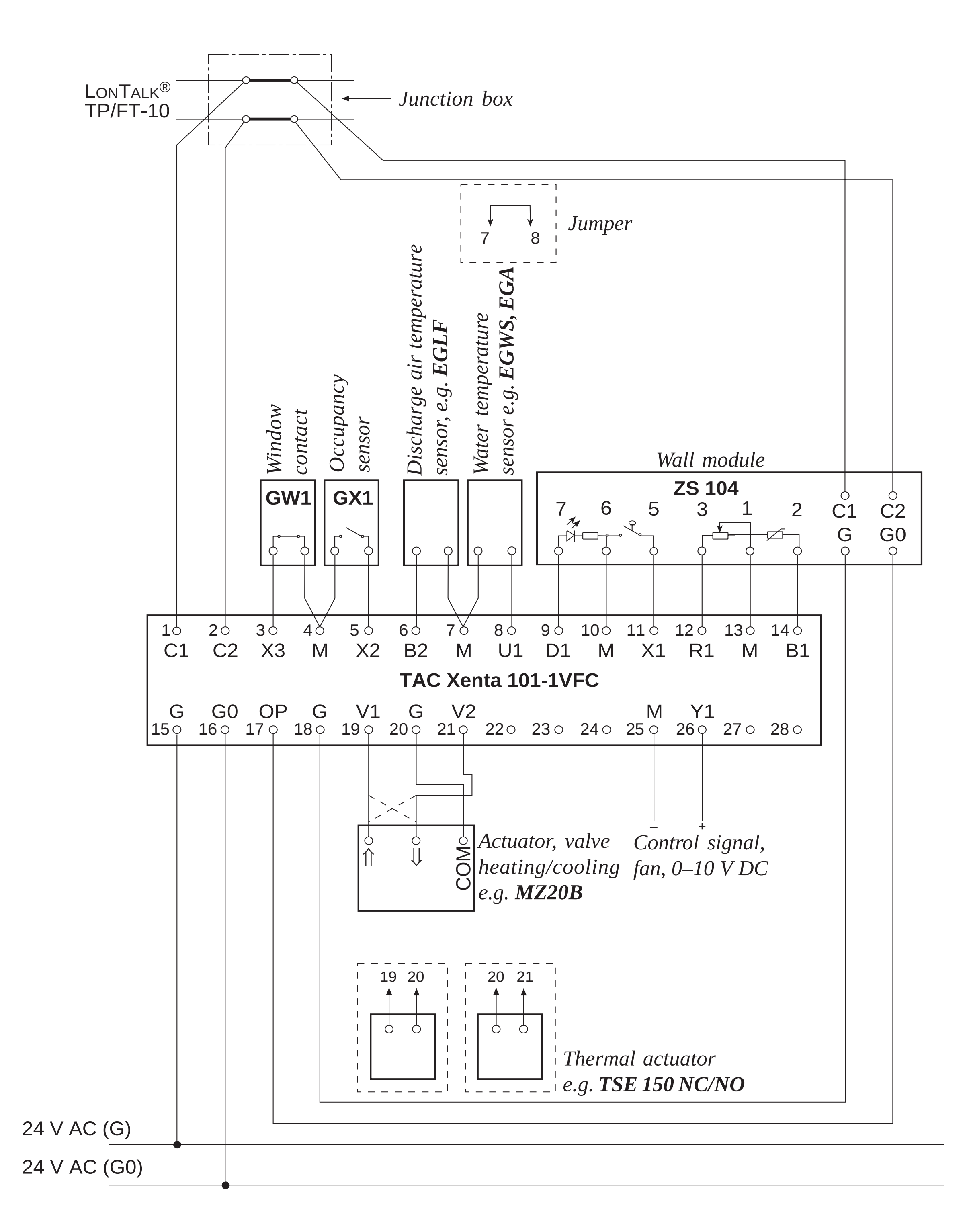


Figure 3.5 Wiring of TAC Xenta 101-1VFC

3.2.4 Wiring of TAC Xenta 101-2VF

Note! Read section 3.2.1 “General” before you connect the cables according to the wiring diagram in figure 3.6.

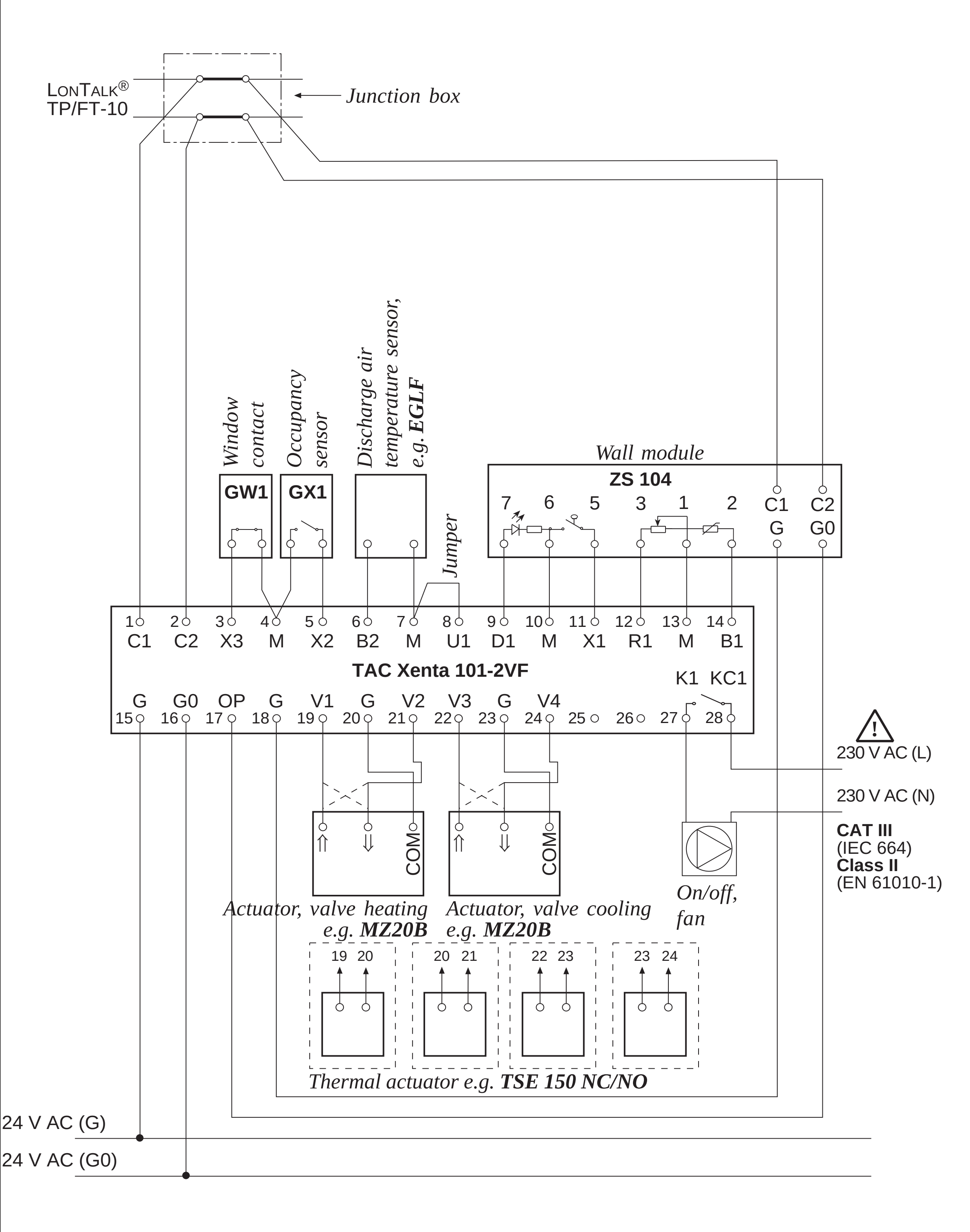


Figure 3.6 Wiring of TAC Xenta 101-2VF

3.2.5 Wiring of TAC Xenta 101-2VFC

Note! Read section 3.2.1 “General” before you connect the cables according to the wiring diagram in figure 3.7.

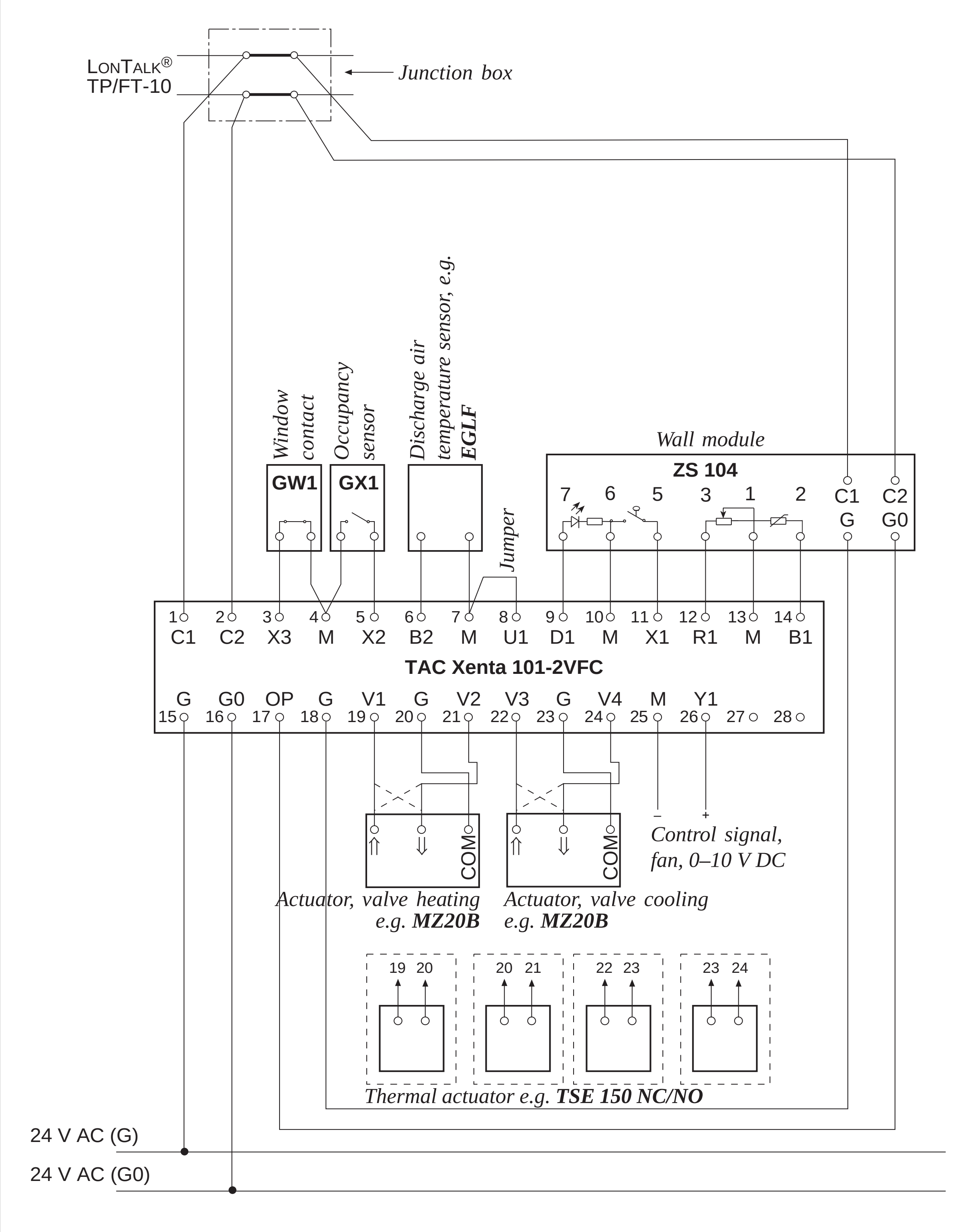


Figure 3.7 Wiring of TAC Xenta 101-2VFC

3.3 Commissioning

3.3.1 General

When the mechanical and electrical installation has been made, you can commission the controller. This means:

- Installing the controller on the network, set node status and give it an address.
- Set the controller's configuration parameters.
- Bind network variables.
- Test the function.

When it comes to commisioning of complete zone systems, read the manual “TAC Xenta - Zone Systems Guideline”. Here you will find a short description of what to do and when to do it.

In short: you could use TAC Xenta OP for setting the basic parameters. Use a network management tool or TAC Vista for commissioning the controller on the network and do the rest of the commissioning.

When TAC Xenta 100 will be used stand-alone, this is how:

1. Set node status to “Configured” with TAC Xenta OP.
2. Set the basic parameters with TAC Xenta OP.
3. Set the other parameters and variables with TAC Xenta OP.

You could also use a network management tool for the commissioning.

3.3.2 Node status

The node status indicates which mode the controller is in, when it comes to network configuration and program. The status can be changed with TAC Vista (version 3.1 or later), network management tool, or, to some extent, TAC Xenta OP. The controller can be in these states:

Unconfigured

The controller is in this state when delivered from the factory. Neither the program nor the network communication are running. The service light emitting diode is flashing.

The controller cannot work on a network in this state. To do so, it must be in configured, online state, see below.

You cannot set configuration parameters or network variables in this state.

Configured, online

By means of TAC Xenta OP, TAC Vista or a network management tool, the status can be changed to configured. Then, both the program and the network communication are running. The service LED is off. This is the normal state for a controller in operation.

Now the controller uses the address which it was given by the tool during configuration. With TAC Xenta OP you cannot, however, set an address. Therefore all controllers get default addresses. This means that such a TAC Xenta 100 cannot work on a network. It can only work stand-alone.

In this state you can set parameters and variables.

Configured, soft online

To get the controller into this state, you need a network management tool. The controller has a program and a network configuration, but the program and the communication are at a standstill. The light emitting diode is off. If the controller is reset, it will go into configured, online.

Configured, hard online

To get the controller into this state, you need a network management tool. The controller has a program and a network configuration, but the program and the communication are at a standstill. The light emitting diode is off. If the controller is reset, it will go remain in this state.

Without a program and not configured

This states indicates that there is something wrong with the controller. No program can be detected. The ligh emitting diode is lit.

3.3.3 Configuration parameters (nci's)

TAC Xenta 100 has a number of configuration parameters, where you can set how the controller should be working. Read about them in chapter 4. There are also network variables which controls the controller during operation.

Use the commissioning protocol in Appendix B to write down your settings at commissioning. In chapter 8, there is information on all parameters and variables, such as their index, accepted values, normal values. There are detailed descriptions of the parameters and variables in chapter 4, 5, and 6.

3.3.4 Network installation

For network installation, you need either a network management tool (LNS based or not) or TAC Vista. Examples of network management tools are MetraVision and ICELAN-G. Here you find brief information on how this is made. You find more information in “TAC Xenta, Guidelines for zone applications”.

The installation has two steps:

1. Feed information about the controllers’ unique neuron-ID into the network management tool’s data base.

2. Let the network management tool install the controller on the network. The controller will then also get an address.

There are two ways to feed the neuron-ID into the data base:

1. Manually feed the neuron-ID into the network management tool. To make this easier you can use a bar code reader to read the detachable ID-neuron label, which you find on every controller. It is suitable to gather these labels when you go around and make the basic configuration, and stick them to a form, drawing or similar. In the manual “TAC Xenta, Guidelines for zone applications” there is a form for this purpose.

2. Use the service pin function. You can only do this when the controller is connected to the network. On the controller there is a service pin key in a hole in the upper left corner, at terminal C1. When you push this, the controller sends out its neuron-ID. The network management tool can then read the neuron-ID from the network, to save it in its data base.

3.3.5 Network variable binding

How binding is done depends on which network management tool is used. To get exact information, you should use the tool’s documentation. In “TAC Xenta Network manual“, there is howevere a description of how network variables are bound with Metra Vision.

To bind network variables is not an issue when the controller is used in stand-alone operation.

3.3.6 Function test

You should also make sure that the control works as intended.

In chapter 5 all the controller’s functions are described.

In chapter 6 you find help, should a problem occur.

Blank page

4 Configuration parameters

All communication with the controller is made by means of network variables. *nci's* are used to configure the controller, *nvi's* controls the controller during operation, and *nvo's* are output variables, which the controller sends out on the network. *nci's* are normally set during commissioning, and are not altered during normal operation (the parameters are stored in a special memory, and can be changed a maximum of 10 000 times). In chapter 8, there is detailed information on accepted values and normal values for all parameters. All configuration parameters have normal values on delivery.

4.1 Basic parameters

nciAppOptions

These parameters are used to set selectable functions in the controller. The parameter consists of 16 bits, where each bit represents one function choice. The bits 10 through 14 are not used. When you look at *nciAppOptions* with TAC Xenta OP, bit 0 is shown to the left.

There is an overview of all the bits' functions in a table below.

Table 4.1 The function of different bits in nciAppOptions.

Bit no.	Function	
Bit 0	0	Occupancy sensor not connected, terminal X2
	1	Occupancy sensor connected, terminal X2
Bit 1	0	Energy hold off device (window contact) not connected, terminal X3
	1	Energy hold off device (window contact) connected, terminal X3
Bit 2	0	Cascade control disabled, no discharge air temp. sensor in terminal B2
	1	Cascade control enabled, discharge air temp. sensor in terminal B2
Bit 3	0	Cooling valve enabled (only types 2VF(C))
	1	Cooling valve disabled (only types 2VF(C))
Bit 4	0	Heating valve enabled (only types 2VF(C))
	1	Heating valve disabled (only types 2VF(C))
Bit 5	0	Thermal actuators normally closed (NC)
	1	Thermal actuators normally open (NO)
Bit 6	0	Actuators are increase/decrease models
	1	Actuators are thermo-actuators
Bit 7	0	Slave mode disabled
	1	Slave mode enabled
Bit 8	0	Occupancy sensor: closed contact indicates occupancy
	1	Occupancy sensor: open contact indicates occupancy
Bit 9	0	If *nviSetpoint* has a valid value, the heating/cooling setpoints for the comfort and economy modes are calculated using method B (see section 5.2.4).
	1	If *nviSetpoint* has a valid value, the heating/cooling setpoints for the comfort and economy modes are calculated using method A (see section 5.2.4).
Bit 15		Reserved for production test. Should not be altered!

Bit 10 through bit 14 are not used.

4.2 Other configuration parameters

The controller's other configuration parameters are listed below together with a short description. See also chapter 8.

Index	Name	Description
0	nciLocation	Location label
25	*nciSetpoints*	Occupancy temperature setpoints
26*1	*nciXoverLimit*	Heating/cooling changeover
27	*nciSpaceTempDev*	Max. deviation of zone temperature
28	*nciSpaceTempLow*	Low limit of zone temperature
29	*nciDischAirMax*	Max. limit discharge air
30	*nciDischAirMin*	Min. limit discharge air
31	*nciGainHeat*	Gain for heating controller
32	*nciItimeHeat*	Integral time heating controller
33	*nciHeatActStTime*	Stroke time for heating actuator
34	*nciGainCool*	Gain for cooling controller
35	*nciItimeCool*	Integral time cooling controller
36	*nciCoolActStTime*	Stroke time for cooling actuator
37*2	*nciMinFanSpeed*	Minimum fan speed
38	*nciSpaceTempOfst*	Zone temperature sensor adjustment
41	*nciHeatPrimMin*	Minimum output heating controller
42	*nciInstallType*	Source for network configuration
43	*nciSndHrtBt*	Send heartbeat
44	*nciRcvHrtBt*	Receive heartbeat

*1 Only controllers 1VF and 1VFC
*2 Only controllers 1VFC and 2VFC

nciLocation

The parameter is used to make a label for the actual place where the controller is installed. In the operating panel, this parameter is shown as the first variable (see section 8.1).

nciSetpoints

The parameter is used for setting the setpoint temperatures for heating and cooling in the different operation modes: comfort, economy and off mode (see section 5.2.1 and 5.2.4).

nciXoverLimit

The parameter is used for setting the temperature limit for the changeover between heating and cooling (see section 5.2). Default value 22 °C.

nciSpaceTempDev

The parameter is used for setting the maximum allowed deviation of the zone temperature (see section 5.2.3). Default value 2 °C.

nciSpaceTempLow

The parameter is used for setting the lowest allowed zone temperature (see section 5.3.6). Default value 10 °C.

nciDischAirMax, nciDischAirMin

The parameters are used for setting the allowed maximum/minimum temperatures when using cascade control (see section 5.3.2). Default values 35/10 °C.

nciGainHeat, nciGainCool

The parameters are used for setting the gain for the heating/cooling controllers. Default value 25.

nciItimeHeat, nciItimeCool

The parameters are used for setting the I-time for the heating/cooling controllers. Default value 900 s (15 min).

nciHeatActStTime, nciCoolActStTime

The parameters are set according to the runtime of the actuator.

nciMinFanSpeed

The parameter is used for setting the minimum fan speed (see section 5.2.7).

nciSpaceTempOfst

The parameter is used for adjusting the temperature setpoints. Default value 0.0%.

nciHeatPrimMin

The parameter is used for setting the minimum value for the opening of the heating valve (see section 5.3.5). Default value 0%.

nciInstallType

The parameter is only used during free-standing operation and is set to show that the node itself should define its address (see section 8.5.3).

nciSndHrtBt

The parameter is used to decide how often the nvo's, which are sent out on the network regularly, should be sent (see section 8.3).

nciRcvHrtBt

The parameter is used to decide the maximum time there can be between updating the nvi's, for which the controller expects continuous updating (see section 8.3).

5 Functional description

5.1 General

The controller’s function is determined by its node status (section 3.3.2), different operations (section 5.2.1) and the ways to force the controller (section 5.2.2) for well-adapted zone temperatur control. The controller has a built-in fan function, measures the zone temperature, and uses different methods to calculate set-points. Apart from the basic functions in chapter 5.2, the controller has a number of other possibilies to control the climate in the zone. There are information about these functions in chapter 5.3.

Each section in this chapter is ended with information on which network variables are used in the current control situation. If you need details about the network variables’ characteristics, such as default values and accepted values, you find this in chapter 8.

5.2 The controller's basic functions

5.2.1 Operation modes

The controller has four selectable operation modes:

- Comfort
- Economy
- Bypass
- Off

The operation mode is controlled by *nviManOccCmd*, but is aslo influenced by occupancy sensors and the bypass key on the wall module. The connection between these is shown in table 5. There you also find the controller's values during stand-alone operation.

Table 5.1 The relation between desired operation, bypass timer, occupancy sensor and current operation mode.

Desired operation ***nviManOccCmd***	**Bypass timer**[1]	**Occupancy sens.**[2]	**Current op. mode**	***nvoEffectOccup***
Comfort	Enabled	Without signific.	Comfort	OC_OCCUPIED
	At a stand-still	Occupancy detect.	Comfort	OC_OCCUPIED
OC_OCCUPIED		No occupancy	Economy	OC_STANDBY
Economy	Enabled	Without signific.	Bypass	OC_BYPASS
OC_STANDBY	At a stand-still	Without signific.	Economy	OC_STANDBY
Off	Enabled	Without signific.	Bypass	OC_BYPASS
OC_UNOCCUPIED	At a stand-still	Without signific.	Off	OC_UNOCCUPIED
Stand-alone	Enabled	Occupancy detect.	Comfort	OC_OCCUPIED
		No occupancy	Bypass	OC_BYPASS
	At a stand-still	Occupancy detect.	Comfort	OC_OCCUPIED
OC_NUL		No occupancy	Off	OC_UNOCCUPIED

[1] Activated by the bypass key on the wall module

[2] See section 5.3.4 about occupancy sensors

Comfort mode

This is the default mode, when someone is in the zone, and the controller should give the room a comfortable climate. The controller is in this mode when *nviManOccCmd*=OC_OCCUPIED (or OC_NUL after a power down).

The LED on the wall module is lit with a steady red light and you can use the setpoint knob on the wall module to make a manual setting. The setpoints used are found in *nciSetpoints* (can be modified).

The fan is on. In controllers 1VFC and 2VFC, the fan speed is controlled, if necessary, to maintain the zone temperature. The alarm for zone temperature deviation can cut out, but the alarms for window contact and low zone temperature are blocked.

Economy mode

In economy mode, the controller lowers the energy consumption in the zone by using the heating and cooling setpoints for economy in *nciSetpoints* (can be modified). The controller is in this mode when *nviManOccCmd* = OC_STANDBY and the bypass key has not been pressed.

The LED of the wall module flashes slowly. The bypass key can be used, and also the setpoint knob, if you want to make a manual setting.

The fan is off in all controllers, if there is no demand for heating or cooling. In such cases, the fan is running. The alarm for zone temperature deviations is blocked, but the alarm for low zone temperature and window contact can cut out.

Bypass mode

The bypass key on the wall module is used if you want to turn to comfort mode occasionally from economy or off mode.

When someone presses the bypass key on the wall module, the bypass timer is started and the controller turns to bypass mode. The bypass timer runs for two hours, and after those two hours the controller changes operation mode according to table 5.1. The controllers bypass mode acts as the comfort mode during those two hours. Both setpoints and alarms work as in comfort mode.

Off mode

When the zone is not used for a longer period of time, the controller can be set in off mode. The controller is in this mode when *nviManOccCmd*=OC_UNOCCUPIED.

The light emitting diode on the wall module is out, and the fan is off in all controllers, if there is no demand for heating or cooling. In such cases, the fan is running. The setpoint knob is blocked, but the bypass key is not. The setpoints used are found in *nciSetpoints*, off mode. The alarm for zone temperature deviations is blocked, but the alarm for low zone temperature and window contact are enabled.

Index	Variable name	Description
1	*nvoEffectOccup*	Actual occupancy output
14	*nviManOccCmd*	Occupancy scheduler input
25	*nciSetpoints*	Occupancy temperature setpoints

5.2.2 Forcing the controller

TAC Xenta 101 is designed to control both heating and cooling, and to automatically change over between heating and cooling.

The 2-valve types changes control sequence by comparing the room temperature with each respective setpoint.

The 1-valve types determines heating/cooling cases by sensing whether the coil gets hot or cool water. Therefore the controller must have a water temperature sensor, to make the changeover automatic.

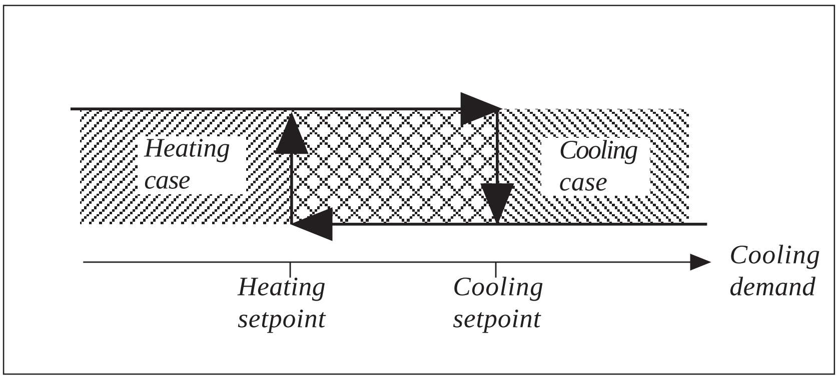


Figure 5.1 Changeover between heating and cooling cases (controllers 2VF and 2VFC)

You can force the controller to heat only or cool only, just as you can force it to neither heat nor cool, and to run the fan only. It is done with *nviApplicMode,* according to the table below.

Table 5.2 The relation between nviApplicMode and forcing.

nviApplicMode	**Forcing**	**Description**
HVAC_AUTO	Automatic (no forcing)	The controller automatically changes over between heating and cooling. 1VF and 1VFC does it by comparing the water temperature with *nciXoverLimit*. 2VF and 2VFC do it by comparing the room temperature with cooling and heating setpoints.
HVAC_HEAT	Heating only	The controller can only heat. The cooling setpoint is neglected, as also any water temperature sensor.
HVAC_COOL	Cooling only	The controller can only cool. The heating setpoint is neglected, as also any water temperature sensor.
HVAC_FAN_ONLY	Fan only	The controller neither cools nor heats, if the frost protection does not cut out. The fan is running, max. speed in 1VFC and 2VFC.
HVAC_OFF	Off	The controller neither cools nor heats, if the frost protection does not cut out. The fan is at a stand-still.

Index	**Variable name**	**Description**
15	*nviApplicMode*	Application mode input
26	*nciXoverLimit*	Heating/cooling changeover

5.2.3 Measuring zone temperature

You can measure the zone temperature either with a permanent thermistor sensor (usually the wall module) or with a LonTalk temperature sensor node connected to *nviSpaceTemp*. If *nviSpace-Temp* has a valid value, the controller will use this, otherwise the thermistor value will be used. The thermistor value (or a value from the network) can be adjusted by *nciSpaceTempOfst* having received a value; this is added to the thermistor value. The value the controller uses is also put out on *nvoSpaceTemp*. If neither value is valid, *nvoSpaceTemp* gets the off value. *nvoSpaceTemp* is sent out when it has changed at least 0,1°C.

Index	Variable name	Description
6	*nvoSpaceTemp*	Zone temperature output
16	*nviSpaceTemp*	Zone temperature input
38	*nciSpaceTempOfst*	Zone temperature sensor adjustment

5.2.4 Setpoint calculation

Zone temperature setpoints

nciSetpoints defines six temperature setpoints; heating setpoint comfort mode, cooling setpoint comfort mode, heating setpoint economy mode, cooling setpoint economy mode, heating setpoint off mode and cooling setpoint off mode.

The smallest accepted deviation between the heating and cooling setpoints is 0,5 °C, and the heating setpoints must be higher than the cooling setpoints. If the heating setpoints are higher or equal to the cooling setpoints, the controller resets the heating setpoint to 0,5 °C lower than the cooling setpoint. Table 5.2 shows accept-ed values and default values for the six temperature setpoints in *nciSetpoints*.

The setpoints for comfort and economy mode are basic setpoints, which can be changed with *nviSetpoint*, *nviSetPntOffset* and the setpoint knob. The off mode setpoints are always valid.

Table 5.2 The setpoints in nciSetpoints.

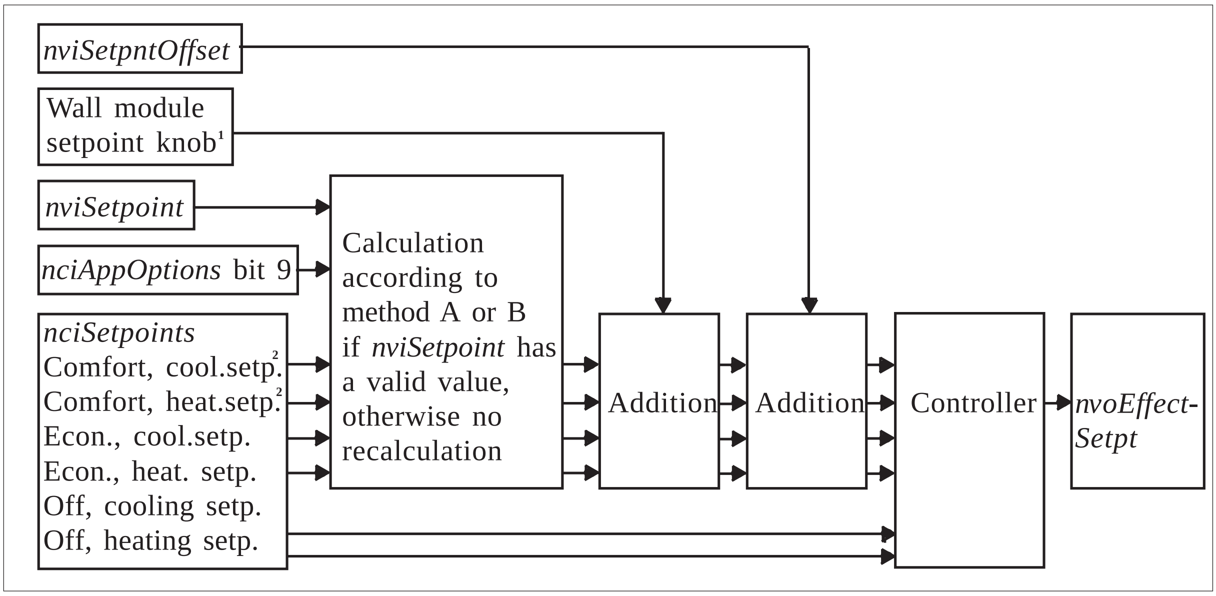
Setpoint	*Min.*	*Max.*	*Normal*
Cooling setpoint comfort	*10 °C*	*35 °C*	*23 °C*
Heating setpoint comfort	*10 °C*[1]	*35 °C*	*21 °C*
Cooling setpoint economy	*10 °C*	*35 °C*	*25 °C*
Heating setpoint economy	*10 °C*[1]	*35 °C*	*19 °C*
Cooling setpoint off	*10 °C*	*35 °C*	*28 °C*
Heating setpoint off	*10 °C*[1]	*35 °C*	*16 °C*

[1]If the cooling setpoint is 10 °C, the heating setpoint is set to 9,5 °C.

Calculation

The current setpoint, *nvoEffectSetpt*, depends on the current operation mode, *nvoUnitStatus*, the desired operation mode, *nviApplicMode*, and *nviSetpoint*, *nviSetpntOffset*, *nciAppOptions*, *nciSetpoints* and a possible local setpoint adjustment via the wall module. Figure 5.2 shows an overview over the relation between the variables used for setpoint calculation.

Figure 5.2 The relationen between variables for the setpoint calculation.



[1] The wall module's setpoint knob only affects comfort and economy mode.
[2] In comfort mode, the setpoints for method A and method B are the same.

nviSetpoint is used to allow the temperature setpoints in comfort and economy mode to be changed via the network. If there is a valid value on *nviSetpoint*, the controller calculates the setpoints for comfort and economy mode with method A or method B (the methods are described in Appendix A). The choice of method is made via *nciAppOptions*, bit 9. If bit 9=0 method B is used, and if 9=1 method A is used. If there is no valid value on *nviSetPoint*, no recalculation of the temperature setpoints in *nciSetpoints* is made.

nviSetPntOffset can be seen as a setpoint adjustment from a wall module connected to the network. Its value is added to setpoints for comfort and economy mode.

In Appendix A there are detailed calculation examples of setpoint calculation.

Index	Variable name	Description
2	*nvoUnitStatus*	Unit status output
5	*nvoEffectSetpt*	Actual setpoint output
15	*nviApplicMode*	Application mode input
17	*nviSetPoint*	Temperature setpoint input
18	*nviSetPntOffset*	Setpoint offset input
24	*nciAppOptions*	Application options
25	*nciSetpoints*	Occupancy temperature setpoints

5.2.5 Control sequence with one valve output (controllers 1VF and 1VFC)

The zone temperature is controlled in one step only, which either heats or cools. Figure 5.3 shows the control sequence:

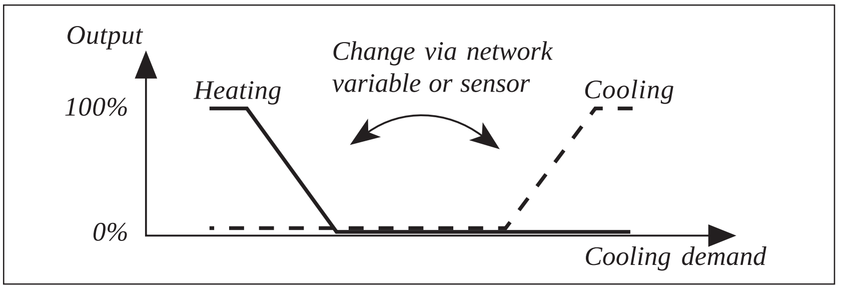


Figure 5.3 Control sequence for heating/cooling coil

How the controller determines whether to heat or cool is shown in the following table.

nviApplicMode	**water temp. > *nciXoverLimit***	**Control case**
HVAC_AUTO	Yes No	Heating Cooling
HVAC_HEAT	No significance	Heating
HVAC_COOL	No significance	Cooling

Table 5.3 The relation between nviApplicMode, water temperature and control case

The window contact overrides the table above and sets the forcing mode to “off”, see chapter 5.3.3.

The water temperature is determined as follows:

If *nviWaterTemp* deviates from the off value, this is used. However, an alarm (bit 13) will cut out if the signal is not within the accepted limits, -10 - + 90.

If *nviWaterTemp* has an off value, the physically connected sensor is used. The changeover between heating and cooling is made with a hysteresis of 2°C of the water temperature. If there is no physically connected sensor, the input should be fitted with a jumper. The cause for this is purely electrical and has nothing to do with the function of the controller.

Valve control is found in section 5.2.6.

Index	**Variable name**	**Description**
15	*nviApplicMode*	Application mode input
19	*nviWaterTemp*	Water temperature input
26	*nciXoverLimit*	Heating/cooling changeover

5.2.6 Control sequence with two valve outputs (controllers 2VF and 2VFC)

The zone temperature is controlled in two steps. The first step is a heating step and the second one a cooling step. Figure 5.4 shows the control sequence.

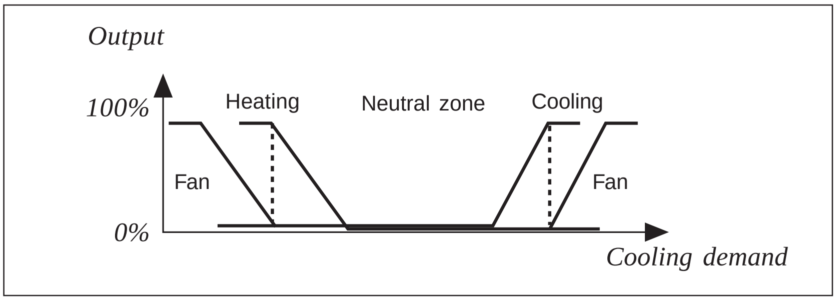


Figure 5.4 Control sequence between heating and cooling coil

The sequence is run as follows: When the cooling demand increases, first the heating valve closes, then the cooling valve opens. When the cooling demand decreases, the sequence is run in the opposite order. As two separate PI-controllers take care of the different steps, you can turn off the heating or cooling sequence with *nviApplicMode* or *nciAppOptions*.

When the valve is completely open, the temperature is controlled with the fan instead. The valve control is made either with an increase/decrease or a thermo actuator. The choice of valve type is made in bit 6 in *nciAppOptions*.

Increase/decrease actuators are controlled with two signals, V1 and V2 for one valve, V3 and V4 for the other.

The thermo actuator is controlled with one signal, V1 for one valve, V3 for the other. However, V2 is set in parallel with V1 and V4 is set in parallel with V3. These can either be used for its own valves, or to supply more current to the regular valves. The actuator type must be “normally closed”, i.e. a set output makes the valve open. The thermo actuator for heating is started when the zone temperature gets below the current setpoint, and is turned off when the zone temperature gets above the current setpoint. The opposite is true for the cooling valve. The changeover is made with a hysteresis of 0,1 °C. When a thermo actuator is used, cascade control (discharge air sensor) cannot be used.

If you have deselected one of the valves in *nciAppOptions*, the controller will not consider its setpoint, no matter what it says in *nviApplicMode*.

Index	Variable name	Description
15	*nviApplicMode*	Application mode input
24	*nciAppOptions*	Application options

5.2.7 Fan control

On/off control

The controllers 1VF and 2VF use a relay output to control the fan on or off. In comfort and bypass mode, the fan runs constantly, and in economy and off mode, the fan is off, if there is no demand for heating or cooling. If necessary, the fan runs.

When the controller is forced into fan mode, the fan always run. When the controller is forced into off mode, the fan is off, apart from when the frost protection has cut out.

Speed control

The controllers 1VFC and 2VFC use output Y1 to control the fan speed. *nciMinFanSpeed* (lowest fan speed) is used to set the fan speed.

Increase/decrease actuator: In comfort and bypass mode, the fan runs at its lowest speed as long as the heating and/or cooling valve can control the zone temperature. When either of the valves is completely open, the controller controls the fan speed to be able to control the temperature better.

Thermo actuator: In comfort and bypass mode, the fan runs when the actuator output is enabled. The speed is controlled.

All actuators: In economy and off mode, the fan is off, if there is no demand for heating or cooling. If necessary, the fan runs at its lowest speed.

When the controller is forced into fan mode, the fan runs at its highest speed. When the controller is forced into off mode, the fan is off, apart from when the frost protection has cut out.

See section 5.3.1 for further description of fan control.

Fan control and monitoring via a network

nvoFanSpeed (current fan speed) and fan_output in *nvoUnitStatus* (unit status) show current fan speed. *nvoFanSpeed* can also remote control a fan via the network.

Index	Variable name	Description
2	*nvoUnitStatus*	Unit status output
12	*nvoFanSpeed*	Fan speed output
37	*nciMinFanSpeed*	Minimum fan speed

5.3 More about functions

5.3.1 Heating and cooling control

The zone temperature is controlled with a heating and cooling controller. In two-step control, the heating and cooling controllers work in sequence. Both can be disabled, not depending on each other, if you set the current network variables (see below).

nvoHeatOutput (output heating) and the heating setpoint in *nvoUnitStatus* (unit status) show the current heating output level. *nvoHeatOutput* can be used to remote control a heating source. Accepted values for these variables range from 0% to 100% of the heating capacity. The value 163,83% is sent as a not valid value to show that the heating step is blocked.

nvoCoolOutput (output cooling) and the cooling value in *nvoUnitStatus* show the current cooling output level. *nvoCoolOutput* can be used to remote control a cooling source. Accepted values for these variables range from 0% to 100% of the cooling capacity. The value 163,83% is sent as a not valid value to show that the cooling step is blocked.

nvoTerminalLoad (current output heating/cooling) shows the current heating or cooling demand for the fan coil, and normally the variable is bound to a energy supply node. Negative values indicate a heating demand and positive values indicate a cooling demand.

Heating and cooling controller	
Type:	PI
Gain:	0-32,75; normal value: 25[1]
I-time:	0-60 minutes; normal 15 minutes[2]
Dead band:	0,2 °C
Run time:[3]	5-600 s; normal 165 s
Control interval:	15 s

[1]*nciGainHeat* is used during heating, *nciGainCool* is used during cooling.

[2]*nciItimeHeat* is used during heating, *nciItimeCool* is used during cooling.

[3]Controllers 1VF(C) use *nciHeatActStTime. nciCoolActStTime* is rejected.

Index	Variable name	Description
2	*nvoUnitStatus*	Unit status output
3	*nvoTerminalLoad*	Heating/cooling demand output
8	*nvoHeatOutput*	Heating control output
10	*nvoCoolOutput*	Cooling control output
31	*nciGainHeat*	Gain for heating controller
32	*nciItimeHeat*	Integral time for heating controller
33	*nciHeatActStTime*	Stroke time for heating actuator
34	*nciGainCool*	Gain for cooling controller
35	*nciItimeCool*	Integral time for cooling controller
36	*nciCoolActStTime*	Stroke time for cooling actuator

5.3.2 Cascade control

Cascade control of the discharge air and zone temperatures can be enabled with the configuration parameter *nciAppOptions* (selectable controller functions). In this case a discharge air temperature sensor should be connected to input B2. When cascade control is enabled, the controller uses an additional PI-controller to calculate the discharge air temperature setpoint based on the current zone temperature.

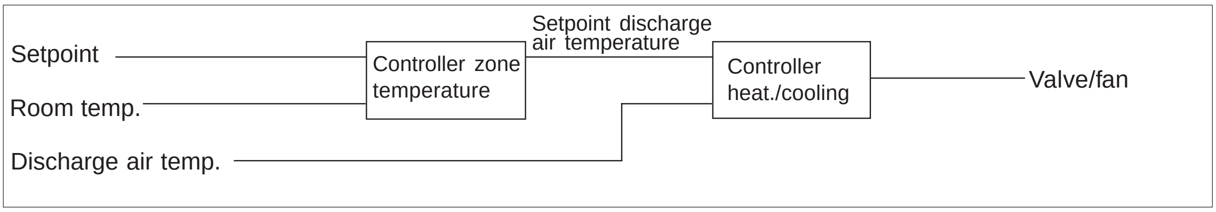


Figure 5.5 Principal diagram cascade control

The discharge air temperature setpoint can be limited to avoid that the controller lets too cool or too hot air into the zone (with *nciDischAirMin* and *nciDischAirMax*). The limits can be disabled by setting them to not valid values.

The controller sends the discharge air temperature in the variable *nvoDischAirTemp*. If the controller does not have a sensor, the not valid value is sent. The variable is sent on the network when its value has changed more than 0,5 °C since the last sendout.

Zone temperature controller	
Type:	PI
Gain:	0–32,75; normal value 25[1]
I-time:	0–60 minutes; normal 15 minutes[2]
Dead band:	0,2 °C
Control interval:	60 s

[1]*nciGainHeat* is used during heating, *nciGainCool* is used during cooling.
[2]*nciItimeHeat* is used during heating, *nciItimeCool* is used during cooling.

Heating and cooling controller	
Type:	PI
Gain:	2
I-time:	3 minutes
Dead band:	0,2 °C
Run time:	5–600 s; normal 165 s
Control interval:	15 s

Index	Variable name	Description
7	*nvoDischAirTemp*	Discharge air temperature output
24	*nciAppOptions*	Application options
29	*nciDischAirMax*	Max. limit discharge air
30	*nciDischAirMin*	Min. limit discharge air

5.3.3 Window contact

TAC Xenta 101 is designed to be able to limit the energy consumption when a window in the room is open. You can connect a local sensor directly to the controller, digital input X3, or use *nviEnergyHoldOff*. The energy hold off is enabled when either of these signals indicate an open window. The energy hold off is made by the controller being set to off mode.

To be able to use a sensor (local or connected to the network), bit 1 in *nciAppOptions* must be set to 1.

nvoEnergyHoldOff has the value of the locally connected sensor. This is true even if bit 1 in *nciAppOptions* is set to 0.

If the energy hold off has been active for 60 sec. the window contact alarm cuts out, bit 2 in *nvoAlarmstatus*.

Index	Variable name	Description
4	*nvoAlarmstatus*	Alarm status output
13	*nvoEnergyHoldOff*	Energy hold off output
20	*nviEnergyHoldOff*	Energy hold off input
24	*nciAppOptions*	Application options

5.3.4 Occupancy sensor

There can be a sensor connected to TAC Xenta 101 to determine whether someone is in the room or not. If there is no occupancy sensor connected, the controller supposes that there is always someone in the room. The controller uses the information to determine whether the operation mode should be comfort or economy. When the controller is used stand-alone, the sensor is used to choose between comfort mode or off mode. See table 5.1 in chapter 5.2.1.

The sensor can be connected either directly to the controller, input X2, or via the network, *nviOccSensor*. To be able to use a directly connected sensor, bit 0 in *nciAppOptions* must be set to 1. When *nviOccSensor* has received a valid value, this is used, whether there is a directly connected sensor or not.

Bit 8 in nciAppOptions indicates whether input X2 should mean presence or absence. Bit 8=0 means that an open input X2 means presence. Bit 8=1 means that a closed input X2 means presence.

The directly connected sensor's value is sent out on the network in *nvoOccSensor*. If there is no sensor connected, (according to nciAppOptions) the value OC_NUL is sent out.

There is a 20 minute delay before the operation mode is changed from comfort to economy in *nvoEffectOccup*. The change in *nvoOccSensor* only takes 250 ms to make other uses of the occupancy sensor possible (lighting, alarm etc.).

Index	Variable name	Description
24	*nciAppOptions*	Application options
39	*nvoOccSensor*	Occupancy sensor output
40	*nviOccSensor*	Occupancy sensor input

5.3.5 Minimum value for heating valve

During cold periods, there is often a back draught at the windows in the room. To avoid this, Xenta 101 offers a possibility to have a little heat on even if it is not really necessary to keep the temperature in the room.

This is done by setting a lowest value for the opening of the heating valve. Xenta 101 makes sure that the opening never falls below this value. The value is given as a percentage in *nciHeatPrimMin* (lowest output heating valve).

The function can only be used when a increase/decrease actuator is used, not when a thermal actuator is used. The function uses the heating valve run time, *nciHeatActStTim*, to control the opening of the valve.

To accomplish this function the controller needs to know the real actuator position. For this purpose a min.-position calibration is performed at certain occasions. This is done by forcing the decrease output during 5 min. allowing the actuator to reach its minimum position. This operation takes place once every day and also at operating mode changes and at a randomly chosen time after power up.

The value of the network variables which show how Xenta 101 sets the heating valve, *nvoUnitStatus* (heat_output_primary), *nvoHeatOutput* and *nvoTerminalLoad*, which shows the position of the valve, should not be lower than *nciHeatPrimMin*.

Index	Variable name	Description
2	*nvoUnitStatus*	Unit status output
3	*nvoTerminalLoad*	Heating/cooling demand output
8	*nvoHeatOutput*	Heating control output
33	*nciHeatActStTime*	Stroke time for heating actuator
41	*nciHeatPrimMin*	Minimum output heating controller

5.3.6 Alarm

When TAC Xenta 101 reports alarms to a monitoring system, this is done with the network variable *nvoAlarmStatus*. The variable has 16 bits, which corresponds to different alarm situations.

Alarm modes for nvoAlarmStatus

Bit no	Alarm	Cuts out when...	Is reset when...
0	Deviating zone temperature	The deviation in zone temp. is larger than *nciSpaceTempDev* for more than 60 minutes (Comfort mode).	The deviation in zone temp. is smaller than the value in *nciSpaceTempDev* (hysteresis 0,5 °C).
1	Low zone temperature	The zone temp. is lower than the value in *nciSpaceTempLow* for more than 60 min. (Economy and off mode).	The zone temp. is more than 2°C above the value in *nciSpaceTempLow.*
2	Window contact alarm	Energy hold off (window contact) is active for more than 60 s (Economy and off mode).	The controller no longer detects the state.
10	Start not bound *nvi:s*	Power on.	When all not bound network variables have been received.
11	Adaptation of thermistor	Internal writing error in the controller memory.	The controller must be replaced.
12	Bound network variables not received	Bound network variables have not been received within set time. *nciRcvHrtBt*	When network variables have been received.
13	Not valid value on input	An input network variable gets outside its accepted values.	The variable gets an accepted value.
14	No application program	No valid application program.	The application program is loaded. Contact the nearest TAC service point.
15	Cannot write to EEPROM	The controller is faulty.	The controller must be exchanged.

Index	Variable name	Description
4	*nvoAlarmStatus*	Alarm status output
27	*nciSpaceTempDev*	Max. deviation of zone temperature
28	*nciSpaceTempLow*	Low limit of zone temperature
44	*nciRcvHrtBt*	Receive heartbeat

5.3.7 Frost protection

The frost protection is active in the forced modes “fan only” and “off ”. If the room temperature falls below 10 °C, e.g. if a window is open, the heating is turned on. The heating is on until the temperature gets above 11,5 °C. The controller uses the setpoint 12 °C.

5.3.8 Master/slave operation

The controller can control a number of slave units, which makes it possible to control several TAC Xenta 101 controllers within the same zone. When bit 7 in *nciAppOptions* is active (=1) the controller works as a slave, otherwise as a master. The slave and the master controller must be of the same type, i.e. the actuator run times must be identical.

The communicating network variables between the master controller and all slave controllers are bound according to figure 5.6. Apart from *nvoUnitStatus*, no other nvo's have reliable values, and these therefore should not be bound to other units. As the heating and cooling valve in the controllers 1VF and 1VFC are combined, *nvoCoolSlaveR* is not used. Only *nvoHeatSlaveR* and *nviFanSlave* are used.

A TAC Xenta 101 working as a slave controller only controls the heating, cooling and the fan according to the values sent by its master controller on the network. It does not consider other input. The forcing mode of the controller which can be read with *nvoUnitStatus*, is set to off for the slave. The frost protection is blocked.

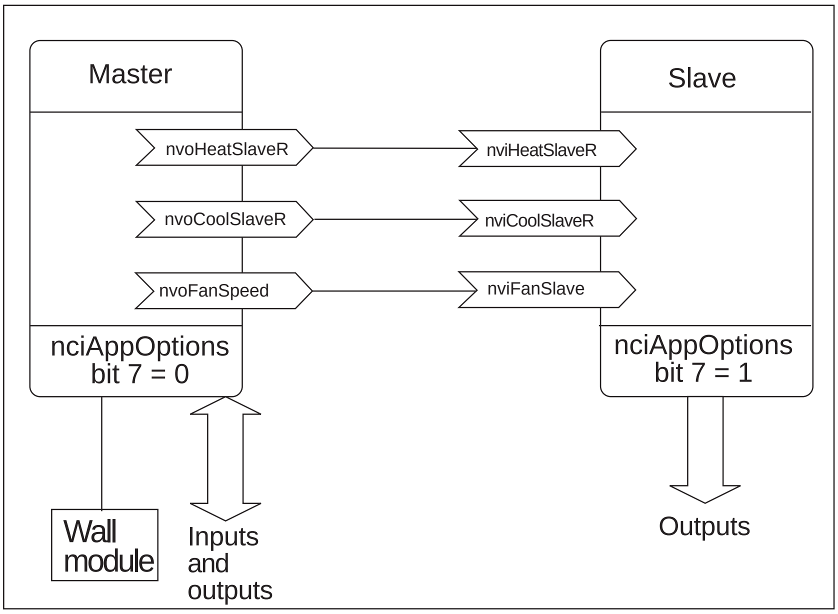


Figure 5.6 Variable bindings between master/slave controllers

Index	Variable name	Description
2	*nvoUnitStatus*	Unit status output
9	*nvoHeatSlaveR*	Heating control output for slave
11	*nvoCoolSlaveR*	Cooling control output for slave
12	*nvoFanSpeed*	Fan speed output
21	*nviHeatSlaveR*	Heating control input for slave
22	*nviCoolSlaveR*	Cooling control input for slave
23	*nviFanSlave*	Fan control input for slave
24	*nciAppOptions*	Application options

Blank page

6 Trouble-shooting

6.1 General

TAC Xenta 101 is normally a very reliable controller. If there are any problems, you can use the trouble-shooting tips in this chapter, preferably when the controller is run on a network, but also when it is used stand-alone. If you need further help, please contact the nearest TAC service point.

6.2 Inputs and outputs (nvi/nvo's)

The most important variables for information on the current status of the controller during operation, are the *nvo's* and the *nvi's*.

With the help of these, you can check the controller's operation and redeem any faults or disturbances.

Below you find the *nvi's* and the *nvo's* with a short description. In chapter 8, you find complete information on all variables' index, variable name, function, accepted values, normal values etc.

Index	Name	Description
1	*nvoEffectOccup*	Effective occupancy output
2	*nvoUnitStatus*	Unit status output
3	*nvoTerminalLoad*	Heating/cooling demand output Positive value=cooling, negative value=heating
4	*nvoAlarmStatus*	Alarm status output (see section 5.3.6)
5	*nvoEffectSetpt*	Effective setpoint output
6	*nvoSpaceTemp*	Zone temperature output, also on input B1
7	*nvoDischAirTemp*	Discharge air temperature output, on input B2
8	*nvoHeatOutput*	Heating control output, corresponding information in *nvoUnitStatus*
9	*nvoHeatSlaveR*	Heating control output for slave
10	*nvoCoolOutput*	Heating control output, corresponding information in *nvoUnitStatus*
11	*nvoCoolSlaveR*	Cooling control output for slave
12	*nvoFanSpeed*	Fan speed output, corresponding information in *nvoUnitStatus*
13	*nvoEnergyHoldOff*	Energy hold off output, window contact status, also on input B1
14	*nviManOccCmd*	Occupancy scheduler input, choice of mode
15	*nviApplicMode*	Application options input (forcing the controller)
16	*nviSpaceTemp*	Zone temperature input, replaces input B1 at a valid value
17	*nviSetpoint*	Temperature setpoint input, which at a valid value, recalculates *nciSetpoints*
18	*nviSetpntOffset*	Setpoint deviation
19	*nviWaterTemp*	Water temperature input, replaces input U1 at a valid value
20	*nviEnergyHoldOff*	Energy hold off input, window contact, determines operation mode together with input X3
21	*nviHeatSlaveR*	Heating control input for slave
22	*nviCoolSlaveR*	Cooling control input for slave
23	*nviFanSlave*	Fan control input for slave
46	*nvoOccSensor*	Occupancy sensor output. Only input X2 is copied. See *nviOccSensor* for net information.
47	*nviOccSensor*	Occupancy sensor input, determines operation mode together with input X2

6.3 Problems and solutions

What affects...	Check...
☞ Operation?	• Bypass timer on wall module (X1). If you have pressed the bypass key, it takes 2 hours before the time expires. • Occupancy sensor (X2) or similar network variable, *nviOccSensor*. If the occupancy sensor has indicated presence, it takes 20 minutes before it is disabled. • How the content in *nvoEffectOccup* can be affected. See section 5.2.1 about operation modes. • Order via network, *nviManOccCmd*.
☞ Operation mode? (Forcing of controller)	• Chosen settings in *nciAppOptions* • Order via network, *nviApplicMode* • If a window contact (X3) or similar network variable, *nviEnergyHoldOff*, is enabled. • Outputs heating/cooling, *nvoUnit-Status*, *nvoTerminalLoad*, *nvoHeatOut-put*, *nvoCoolOutput* which can be affected by normal control or *nciHeatPrimMin*. • The fan speed, *nvoUnitStatus*, *nvo-FanSpeed* which can be affected by normal control or *nciMinFanSpeed*.
☞ Control setpoint?	• Current operation mode, *nvo-EffectOccup* • Current unit status, *nvoUnitStatus* • Set basic setpoints, *nciSetpoints*. Chosen options in *nciAppOptions* concerning calculation method A or B together with *nviSetpoint* control this. Not a valid value in *nviSetpoint* gives the basic setpoints. See section 5.2.4 on setpoint calculation. • *nviSetpntOffset* and/or the setpoint knob on the wall module. These give +/- influence.
☞ Read room temperature?	• Physical reading (B1) or similar network variable, *nviSpaceTemp*. A valid value on the network overrides a physical reading. *nciSpaceTempOfst* can displace the value.

cont.

What affects...	*Check...*
☞ Read discharge air temperature?	• Only physical input (B2). This is used for cascade control according to choice with *nciAppOptions*. The control is max. and min. limited via *nciDischAirMax* and *nciDischAirMin*. See section 5.3.2 on cascade control.
☞ That an alarm is set?	• Current operation mode, *nvoEffect-Occup* • Current values in *nciSpaceTempDev* and *nciSpaceTempLow*. • If a window is open (window contact). See also section 5.3.6 on alarms.
☞ The LED on the wall module?	• That the controller receives power also when the LED is out. • The controller when the service LED is lit. This indicates that the controller does not work correctly and should be replaced. • The controller when the service LED is lit for 15 seconds and the goes out. This is not a fault, but an indication that the controller answers a “wink” command from the network. • Current operation.

7 Technical data

7.1 Technical data

Part number:	
TAC Xenta 101-1VF	0-073-0501
TAC Xenta 101-1VFC	0-073-0502
TAC Xenta 101-2VF	0-073-0503
TAC Xenta 101-2VFC	0-073-0504
Power supply	24 V AC –10% +20%, 50–60 Hz
Power consumption:	
Controller with TAC Xenta OP	4 VA
Digital outputs for -1VF, -1VFC	max. 2×19 VA = 38 VA
Digital outputs for -2VF, -2VFC	max. 4×19 VA = 76 VA
Total, -1VF, -1VFC	max. 42 VA
Total, -2VF, -2VFC	max. 80 VA
Ambient temperature:	
Operation	0 °C – +50 °C
Storage	–20 °C – +50 °C
Humidity	max. 90% RH, non-condensating
Housing:	
Material	ABS/PC-plastic
Protection	IP 30
Colour	grey/red
Dimensions, with semi-protection	127×126×50 mm
without semi-protection	122×126×50 mm
Weight	0,4 kg
Inputs X2–X3 for occupancy sensor and window contact:	
Voltage open contact	23 V DC ± 1 V DC
Current closed contact	4 mA
Min. pulse width X2/X3	250 ms/ 16 s
Outputs V1–V2, all models, and V3–V4, only for -2VF(C), for valve actuators (triac):	
Type of actuator	increase/decrease or thermal NC/NO
Min. output voltage	power supply – 1,5 V
Max. load	0,8 A

Relay output for on-off control of fan, only for -1VF and -2VF, K1 and KC1:
Max. voltage .... 250 V AC
Max. load .... 2 A

Output Y1 for fan control, only -1VFC and -2VFC:
Output voltage range .... 0–10 V DC
Max. current .... 2 mA
Accuracy .... ±0,2 V

Input X1, bypass key on wall module:
Min. pulse width .... 250 ms

Inputs for water, and discharge air temperature sensors, and wall module, U1, only -1VF(C), and B1–B2:
Thermistor type .... NTC, 1800 Ω at 25 °C
Measuring range .... –10 °C – +50 °C
Accuracy .... ±0,2 °C

Input R1, setpoint control in wall module:
Type .... 10 kΩ linear potentiometer
Adjustment range .... –5 °C – +5 °C
Accuracy .... ±0,1 °C

Application program:
Cycle time .... 16 s

LED (light emitting diode) colour:
Voltage supply .... green
Service .... red

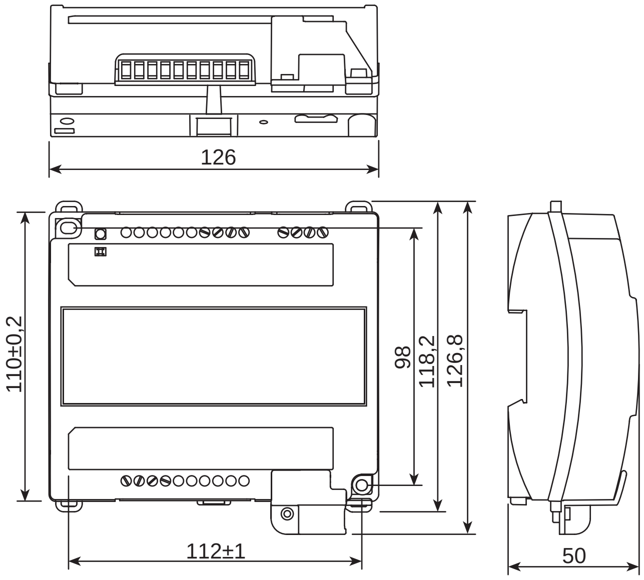
Compatibility:
Standard .... complies with LONMARK® *Interoperability Guidelines* and LONMARK *Functional Profile: Fan Coil Unit*
Network protocol .... LONTALK®
Channel .... TP/FT-10, 78 kbps
Neuron® type .... 3150®, 10 MHz

Standards/Norms:
Emission .... EN 50081-1
Immunity .... EN 50082-1
Safety .... EN 61010-1
ETL listing .... UL 3111-1, first version
.... CAN/CSA C22.2 No. 1010.1-92
Flammability, integrated materials .... UL 94 V-0
CE mark .... complies with the demands

Wall modules:
ZS 101 0-073-0908
ZS 102 0-073-0909
ZS 103 0-073-0910
ZS 104 0-073-0911
Accessories:
Terminal kit, TAC Xenta 100 0-073-0914
Diskett with external interface files (XIF) for TAC Xenta 100 series 0-008-5582

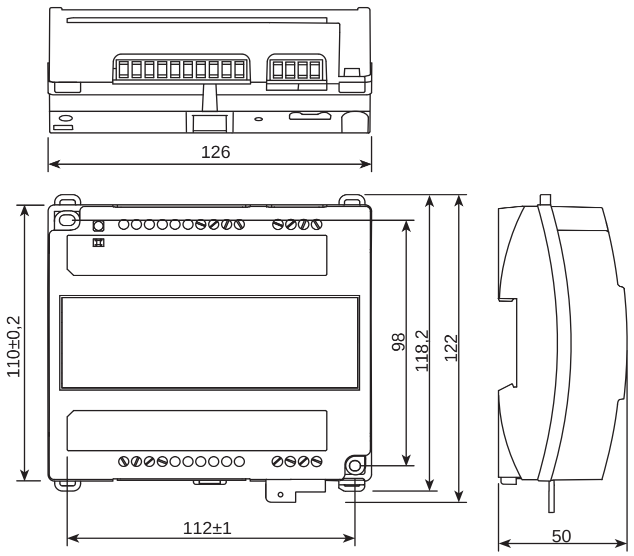
7.2 Dimensions

7.2.1 With semi-protection



Dimensions (mm) for TAC Xenta 101 with semi-protection

7.2.2 Without semi-protection



Dimensions (mm) for TAC Xenta 101 without semi-protection

Blank page

8 Communication

8.1 General

The controller consists of two LonMark™ objects: the node object (section 8.5) and the controller object (section 8.6). These objects are monitored by means of the network variables *nviRequest* (object request) and *nvoStatus* (object status).

The network variable *nciLocation* (position mark) is used when configuring the basic parameters (section 4.1) to give a detailed description of the actual place where the controller is fitted. The variable receives an arbitrary string of signs and dividers as long as the string is no longer than 30 signs. You can program a certain location label, e.g.

```
TAMF.main.floor3.room343/RC40
```

A LNS based network management tool uses *nciLocation* when a data base should be recreated. The monitoring of an already installed network is made by the LNS tool reading *nciLocation*, and then using the information to give the node a subsystem name and a unit name. The string should therefore consist of a name and a search path for the subsystem, followed by a slash and the unit name, i.e.

```
system.subsystem[.subsystem...]/unit name
```

8.2 Default settings and power on

For all network variables the following settings are valid:

- Number of sent messages per time unit: NONE
- Service type: NOT CONFIRMED if not stated otherwise
- Access check: NO, possible to configure: YES.
- Polled: NO for all *nvo* and *nci*, YES for all *nvi* (startingup)
- Synchronized: NO
- Change/update only when the controller is not active on the network; flags = NO
- Restart of TAC Xenta 101 after change; flags = YES

All network variables have the same index as they have in the menu tree in the operator panel TAC Xenta OP. They represent the order, in which they have been declared in the system program, as the order is important for variables' self documentatory string. The variables are of standard type or so called SNVT, and the values each SNVT can receive, are listed in the tables in this chapter. Apart from SNVT, there are also standard configuration parameters (SCPT) and parameter types for user configuration (UCPT). If you want general information about which SNVT/SCPT/UCPT there are and which values they can receive, the “The SNVT Master List and Programmer's Guide” on the internet address www.lonmark.org are a good source of information.

At power on, all variables for inputs and outputs (*nvi* and *nvo*) receives their default values after a restart, as the configuration parameters (*nci*) keep their earlier set values. After a restart all *nvi’s* will send a request to the *nvo’s* they are bound to (a poll).

8.3 Monitoring network variables, Heartbeat

In Xenta 101 there is a function, called Heartbeat, which can be configured to monitor input and output variables on the network.

In the overviews in this chapter, you can see whether the variable is monitored with Heartbeat in the column Hb.

Inputs

Some of the inputs in Xenta 101 are monitored in a way that the variable must receive values within a certain time for it to be regarded as valid. If no value is received within this time, the variable will return to its default value. Also, an alarm is enabled, bit 12 in *nvoAlarmStatus*.

Which outputs are monitored in this way, you find in the list of network variables in chapter 8.6.1.

The time is set with the variable *nciRcvHrtBt*. Its default value is 0.0, which means that no monitoring is performed.

Outputs

The bound outputs are normally sent out when they are changed. Most outputs in Xenta 101 are monitored, so even if the values are not changed, they are sent out at even intervals.

Which outputs are monitored in this way, you find in the list of network variables in chapter 8.6.2.

The time is set with the variable *nciSndHrtBt*. Its normal value is 0.0, which means that no monitoring is performed.

8.4 Not accepted values

All *nvo's* are limited to their accepted values, an all *nvi's* detect whether the incoming value is within the accepted limits. If the value is not accepted, the controller activates bit 13 in the variable for alarm handling, *nvoAlarmStatus*. For a *nvi*, the controller uses the off value, which is also counted as an accepted value.

8.5 The node object

The variables in the node object (figure 8.1) are separated into three categories:

- Mandatory (M)
- Optional (O)
- Configuration properties (C)

The category "Mandatory" contains all compulsory variables*, "Optional" contains selectable variables, and "Configuration properties" contains the configuration parameters.

Note! The network variables' indeces are not the same as the figure in "nv" in the figure.

* According to LonMark standardised function profil for VAV controllers

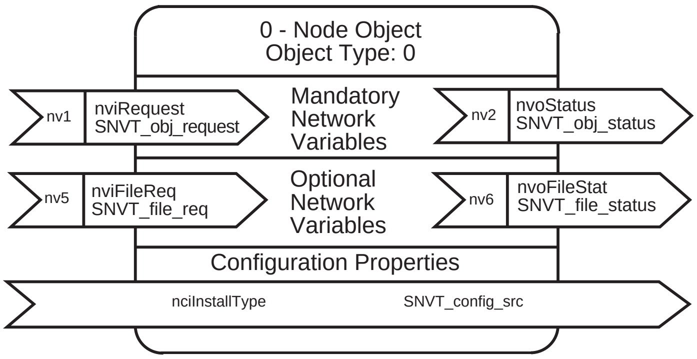


Figure 8.1 The node object

8.5.1 The node object's inputs (nvi)

Index	Variable	Hb*1	SNVT	Accepted values (Service type)	Default value	Description (self doc. string)
45	nviRequest	No	SNVT_obj_request	0=RQ_NORMAL 2=RQ_UPDATE_STATUS 5=RQ_REPORT_MASK	RQ_NUL (Confirmed)	Object request @0|1
47	nviFileReq	No	SNVT_file_req	see "SNVT Master List"	FR_NUL (Confirmed)	File request @0|5

*1Hb=Heartbeat

8.5.2 The node object's outputs (nvo)

Index	Variable	Hb*1	SNVT	Accepted values (Service type)	Default value	Description (self doc. string)
46	nvoStatus	No	SNVT_obj_status	invalid_id (0..1) invalid_request(0..1)	Alla = 0 (Confirmed)	Object status @0|2
48	nvoFileStat	Yes	SNVT_file_status	see ”SNVT Master List”	FS_NUL (Confirmed)	File status @0|6

*1 Hb=Heartbeat

8.5.3 The node object's configuration parameters (nci)

Index	Variable	Hb*1	SNVT SCPT/UCPT	Accepted values	Default value	Description (self doc. string)
42	nciInstallType	No	SNVT_config_src SCPT_nwrk_config (25)	0=CFG_LOCAL 1=CFG_EXTERNAL CFG_NUL	0=CFG_LOCAL	Network configu-ration source &0,,0\x80,25

*1 Hb=Heartbeat

8.6 The controller object

The variables in the controller object (figure 8.2) are separated into four categories:

* According to LonMark standardised function profile for VAV controllers

- Mandatory (M)
- Optional (O)
- Configuration properties (C)
- Manufacturer Defined Section (MDS)

The category “Mandatory” contains all compulsory variables*, “Optional” contains selectable variables, ”Configuration properties” contains configuration parameters, and “Manufacturer Defined Section” is all other variables which make the controller's functions possible.

Figure 8.2 is on the next page. *Note*! The network variables' indeces are not the same as the figure in “nv” in the figure.

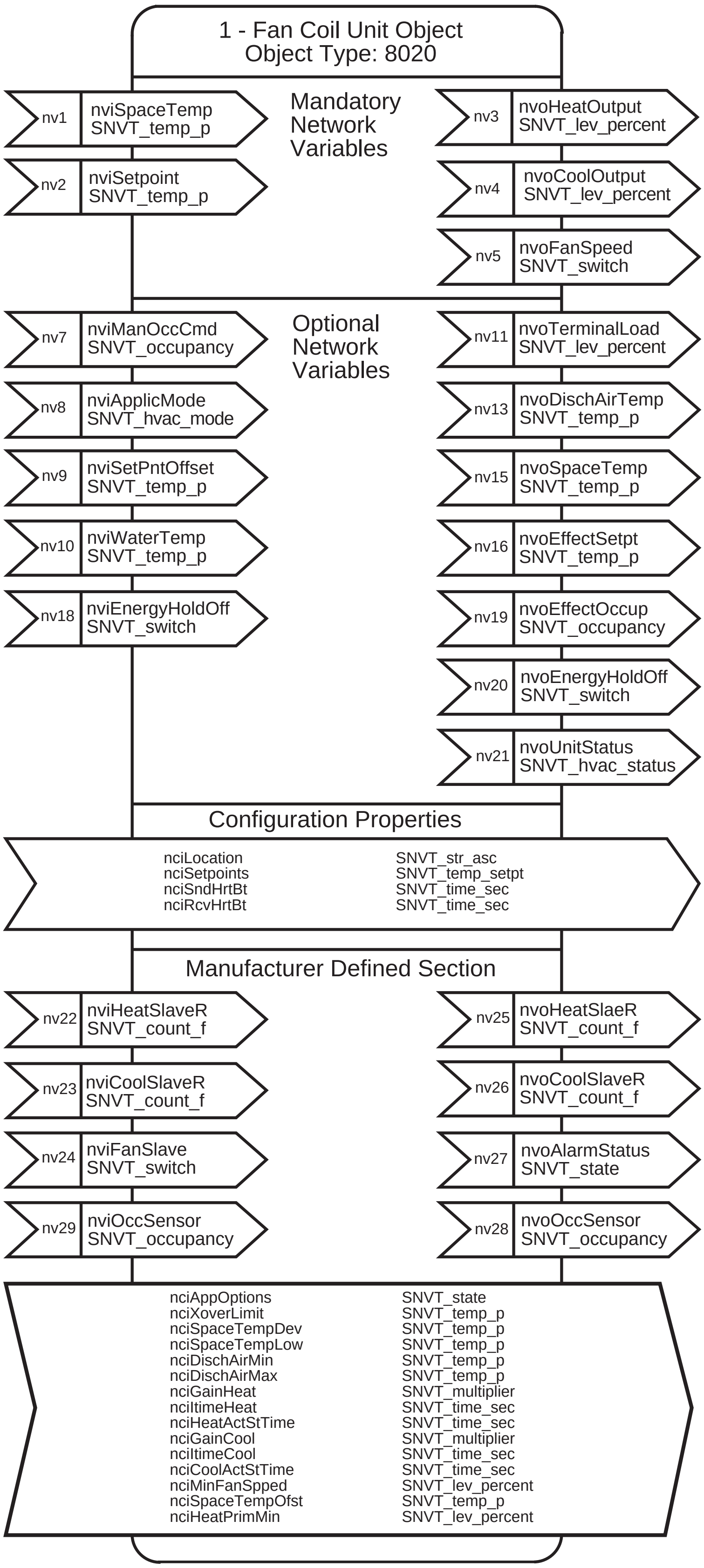


Figure 8.2 The controller object

8.6.1 The controller object's inputs (nvi)

Index	Variable	Hb*1	SNVT	Accepted values	Default value	description (Self doc. string)
14	nviManOccCmd	No	SNVT_occupancy	0=OC_OCCUPIED 1=OC_UNOCCUPIED 3=OC_STANDBY other values=OC_NUL 255=OC_NUL	OC_NUL	Occupancy scheduler input @1|7
15	nviApplicMode	Yes	SNVT_hvac_mode	0=HVAC_AUTO 1=HVAC_HEAT 3=HVAC_COOL 6=HVAC_OFF 9=HVAC_FAN_ONLY; all other values are interpretated as HVAC_AUTO	HVAC_AUTO	Application mode input @1|8
16	nviSpaceTemp	Yes	SNVT_temp_p	–10 °C to 50 °C, 327,67 °C(*2)	327,67 °C(*2)	Zone temperature input @1|1
17	nviSetpoint	No	SNVT_temp_p	10 °C to 35 °C, 327, 67 °C(*2)	327,67 °C(*2)	Temperature setpoint input @1|2
18	nviSetpntOffset	Yes	SNVT_temp_p	–10 °C to 10 °C	0 °C	Setpoint offset input @1|9
19	nviWaterTemp	No	SNVT_temp_p	–10 °C to 90 °C, 327, 67 °C(*2)	327,67 °C(*2)	Water temperature input, @1|10
20	nviEnergyHold Off	Yes	SNVT_switch	0=Off, 1=On, 0% to 100%	Off, 0%	Energy hold off input Enabled at 1=On and ≠ 0%, @1|18
21	nviHeatSlaveR	No	SNVT_count_f	–50 to 50	0	Heating control input for slave, @1#1
22	nviCoolSlaveR	No	SNVT_count_f	–50 to 50	0	Cooling control input for slave, @1#2
23	nviFanSlave	No	SNVT_switch	0=Off, 1=On 0% to 100%	Off, 0%	Fan control input for slave, @1#1
40	nviOccSensor	No	SNVT_occupancy	0=OC_OCCUPIED 1=OC_UNOCCUPIED other values=OC_NUL	OC_NUL	Occupancy sensor input @1#29

*1 Hb=Heartbeat
*2 Off value

8.6.2 The controller object's outputs (nvo)

Index	Variable	Hb*1	SNVT	Accepted values	Default value	Description (Self doc. string)
1	nvoEffectOccup	Yes	SNVT_occupancy	0=OC_OCCUPIED 1=OC_UNOCCUPIED 2=OC_BYPASS 3=OC_STANDBY 255=OC_NUL	OC_OCCUPIED	Actual occupancy output, @1|19
2	nvoUnitStatus	Yes	SNVT_hvac_status mode heat_output_primary heat_output_secondary cool_output econ_output fan_output in_alarm	1=HVAC_HEAT 3=HVAC_COOL 9=HVAC_FAN_ONLY 6=HVAC_OFF 0% to 100% 163,83%(*2) 163,83%(*2) 0% to 100% 163,83%(*2) 163,83%(*2) 0% to 100%, 163,83%(*2) 255(*2)	HVAC_HEAT 163,83%(*2) 163,83%(*2) 163,83%(*2) 163,83%(*2) 163,83%(*2) 255(*2)	Unit status, output, @1|21
3	nvoTerminal Load	Yes	SNVT_lev_percent	–163,84% to 163,84%	0%	Heat./cool. demand output, @1|11
4	nvoAlarmStatus	No	SNVT_state	16 bits, 0=normal, 1 = alarm	00000000 00000000	Alarm status, output, @1#6
5	nvoEffectSetpt	Yes	SNVT_temp_p	10 °C to 35 °C 327,67 °C(*2)	327,67 °C(*2)	Actual setpoint output,@1|16
6	nvoSpaceTemp	Yes	SNVT_temp_p	–10 °C to 50 °C, 327,67 °C(*2)	327,67 °C(*2)	Zone temp. output @1|15
7	nvoDischAir Temp	No	SNVT_temp_p	–10 °C to 50 °C, 327,67 °C(*2)	327,67 °C(*2)	Discharge air temp. output,@1|13
8	nvoHeatOutput	Yes	SNVT_lev_percent	0% to 100%	0%	Heating control output,@1|3
9	nvoHeatSlaveR	No	SNVT_count_f	–50 to 50	0	Heating control output for slave, @1#4
10	nvoCoolOutput	Yes	SNVT_lev_percent	0% to 100%, 163,83%(*2)	163,83%(*2)	Cooling control output, @1|4
11	nvoCoolSlaveR	No	SNVT_count_f	–50 to 50	0	Cooling control output for slave; @1#5
12	nvoFanSpeed	Yes	SNVT_switch	0=Off, 1=On 0% to 100%	Off, 0%	Fan speed output @1|5
13	nvoEnergy HoldOff	Yes	SNVT_switch	0=Off, 1=On 0% to 100%	Off, 0%	Energy hold off output. Off, 0% = no hold off, @1|20
39	nvoOccSensor	No	SNVT_occupancy	0=OC_OCCUPIED 1=OC_UNOCCUPIED 255=OC_NUL	OC_NUL	Occupancy sensor output, @1#28

*1 Hb=Heartbeat
*2 Off value

8.6.3 The controller object's configuration parameters (nci)

Index	Variable	Hb*1	SNVT SCPT/UCPT	Accepted values	Default value	Description (Self doc. string)
0	nciLocation	No	SNVT_str_asc SCPT_location (17)	31 ASCII signs	All = 0	Location label &1,1,0\x80,17
24	nciAppOptions	No	SNVT_state UCPT (1)	16 bits, 0–1	00000000 00000000	Application options &1,1,3\x8A,1
25	nciSetpoints	No	SNVT_temp_setpt SCPTsetPnts (60)	10 °C to 35 °C	occ cool = 23 °C stby cool = 25°C unoc cool = 28°C occ heat = 21°C stby heat = 19°C unoc heat = 16°C	Occupancy temperature setpoints &1,1,0\x80,60, 10:35|10:35|10:35| 10:35|10:35|10:35
26	nciXoverLimit	No	SNVT_temp_p UCPT (12)	0°C to 40°C	22 °C	Heating/cooling changeover &1,1,3\x80,12,0:40
27	nciSpaceTemp Dev	No	SNVT_temp_p UCPT (16)	0°C to 10°C	2 °C	Max.dev.of zone temp. &1,1,3\x80,16,0:10
28	nciSpaceTemp Low	No	SNVT_temp_p UCPT (17)	0 °C to 20 °C	10 °C	Low lim. of zone temp. &1,1,3\x80,17,0:20
29	nciDischAirMax	No	SNVT_temp_p UCPT (18)	0 °C to 40 °C	35 °C	Max. limit disch. air &1,1,3\x80,18,0:40
30	nciDischAirMin	No	SNVT_temp_p UCPT (19)	0 °C to 40 °C	10 °C	Min. limit disch. air &1,1,3\x80,19,0:40
31	nciGainHeat	No	SNVT_multiplier	0 to 32,7675 UCPT (2)	25	Gain for heating controller, &1,1,3\x80,2
32	nciItimeHeat	No	SNVT_time_sec UCPT (3)	0 s to 3600 s (60 minutes)	900 s (15 minutes)	Integral time heating controller &1,1,3\x80,3,0:3600
33	nciHeatActSt Time	No	SNVT_time_sec UCPT (4)	5 s to 600 s (10 minutes)	165 s	Stroke time for heating actuator &1,1,3\x80,4,5:600
34	nciGainCool	No	SNVT_multiplier UCPT (5)	0 to 32,7675	25	Gain for cooling controller. &1,1,3\x80,5
35	nciItimeCool	No	SNVT_time_sec UCPT (6)	0 s to 3600 s (60 minutes)	900 s (15 minutes)	Integral time cooling controller &1,1,3\x80,6,0:3600
36	nciCoolActSt Time	No	SNVT_time_sec UCPT (13)	5 s to 600 s (10 minutes)	165 s	Stroke time for cooling actuator &1,1,3\x80,13,5:600
37*2	nciMinFanSpeed	No	SNVT_lev_percent UCPT (14)	0% to 100%	0%	Minimum fan speed &1,1,3\x80,14,0:100
38	nciSpaceTemp Ofst	No	SNVT_temp_p UCPT (20)	-10,0°C to 10,0°C	0,0 °C	Zone temperature sensor adjustment &2,16,3\x80, 20–10.0:10.0
41	nciHeatPrimMin	No	SNVT_lev_percent	0% to 100%	0%	Min. output heating controller &1,1,3\80,23,0:100
43	nciSndHrtBt	Yes	SNVT_time_sec SCPTmaxSend Time (49)	5,0 s to 6553,4 s 0,0 s = disabled	0,0 s (disabled)	Send heartbeat &2,1.2.3.5.6.8.10.12. 13.34,0\x8A,49
44	nciRcvHrtBt	No	SNVT_time_sec SCPTmaxRcvTime (48)	0,0 s to 6553,4 s 0,0 s = disabled	0,0 s (disabled)	Receive heartbeat &2,15.16.18.20.40, 0\x8A,48

*1 Hb=Heartbeat
*2 Only controllers 1VFC and 2VFC

Appendix A: Setpoint calculation

Definitions:

Deadband	=	Neutral zone
Occupied	=	Comfort mode
Standby	=	Economy mode
Unoccupied	=	Off mode
nviSetPoint	=	Input temperature setpoint (*nviSetpoint*)
nciSetPoints	=	Six basic setpoints for temperature (*nciSetpoints*)
Occupied_cool	=	Cooling setpoint comfort
Standby_cool	=	Cooling setpoint economy
Unoccupied_cool	=	Cooling setpoint off
Occupied_heat	=	Heating setpoint comfort
Standby_heat	=	Heating setpoint economy
Unoccupied_heat	=	Heating setpoint off
Effective	=	Existing

In *nciSetpoints,* the cooling and heating setpoints for comfort and economy mode are set. *nviSetpoint* gives you the possibility to move all four setpoints with only one value. The mean value of the comfort setpoints in *nciSetpoints* can be seen as the basic setpoint for comfort mode, and the mean value of the economy setpoints can be seen as the basic setpoint for economy mode. The temperature scale for the setpoints must be as follows:

unoccupied_heat ≤ standby_heat ≤ occupied_heat ≤ occupied_cool ≤ standby_cool ≤ unoccupied_cool.

There are two methods to calculate the setpoints: Method A and Method B.

Method A:

When *nviSetpoint* receives a valid setpoint, this value becomes the new, common setpoint. The cooling and heating setpoints are recalculated to be at the same distance from the new, basic setpoint as they were from the earlier basic setpoint. Therefore, method A removes the existing asymmetry (see the example on the next page).

The controller calculates the different setpoints for heating and cooling in comfort and economy mode, from *nviSetpoint*, plus or minus half the neutral zone in the comfort and economy modes, which are calculated from the *nciSetpoints*. The controller takes the two heating and cooling setpoints in off mode from *nciSetpoints*.

deadband_occupied = occupied_cool – occupied_heat
deadband_standby = standby_cool – standby_heat

effective_occupied_cool = nviSetPoint + 0,5 (deadband_occupied)
effective_occupied_heat = nviSetPoint – 0,5 (deadband_occupied)
effective_standby_cool = nviSetPoint + 0,5 (deadband_standby)
effective_standby_heat = nviSetPoint – 0,5 (deadband_standby)

Method B:

In economy mode you can chose method B to calculate the existing setpoints. In this case, the setpoints' distance from the existing setpoint, is as far as the distance they had from the old, basic setpoint in comfort mode. Method B only has influence when the two setpoints from *nciSetpoints* do not have the same value, i.e. when the four setpoints are not placed symmetrically around one value. With Method B the asymmetry is therefore kept, as the old comfort setpoint is used (see the example on the next page).

The controller calculates the different setpoints for heating an cooling in comfort and economy modes from *nciSetpoints*. Also, the actual, absolute setpoint deviation is calculated as the mean value of the occupied_heat setpoint and the occupied_cool setpoint. The controller gets the different heating and cooling setpoints in off mode from *nciSetpoints*.

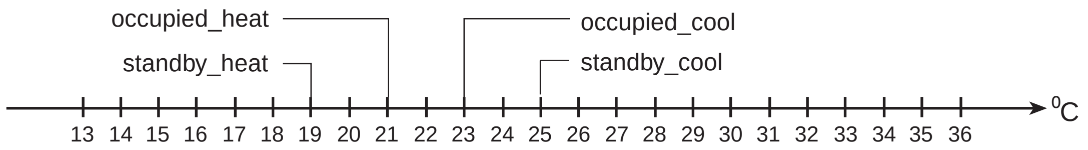
effect_abs_setpoint_ offset = nviSetpoint – (occupied_cool + occupied_heat) /2

effective_occupied_cool = occupied_cool + effect_abs_setpoint_offset
effective_occupied_heat = occupied_heat + effect_abs_setpoint_offset
effective_standby_cool = standby_cool + effect_abs_setpoint_offset
effective_standby_heat = standby_heat + effect_abs_setpoint_offset

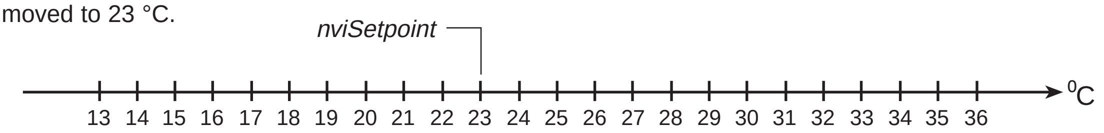
The following two examples show which influence *nviSetpoint* has and what Method A and Method B each means.

Example 1.

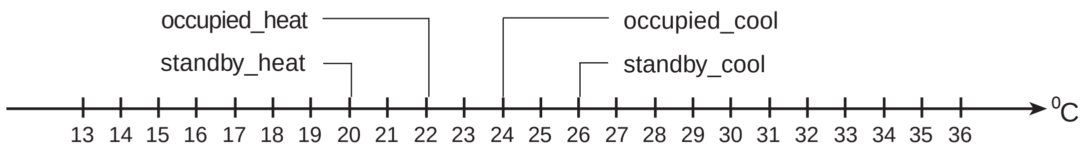
1) Assume that you have the following temperatures in *nciSetpoints*:



2) The basic setpoint for both comfort and economy modes is 22 °C. By means of *nviSetpoint* this is moved to 23 °C.

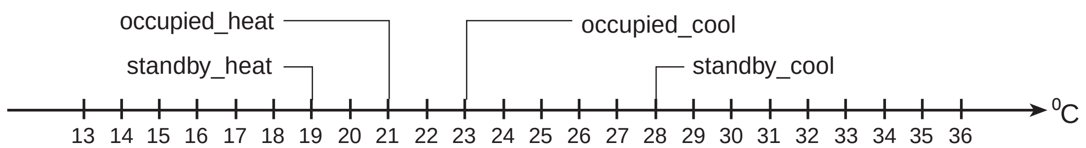


3) With Method A and Method B you then get the following result:

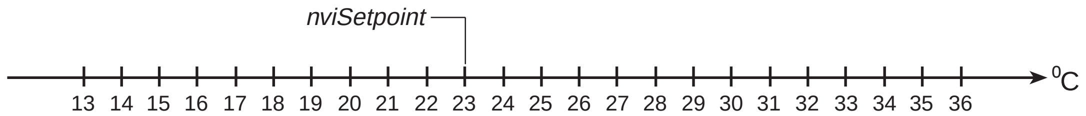


Example 2.

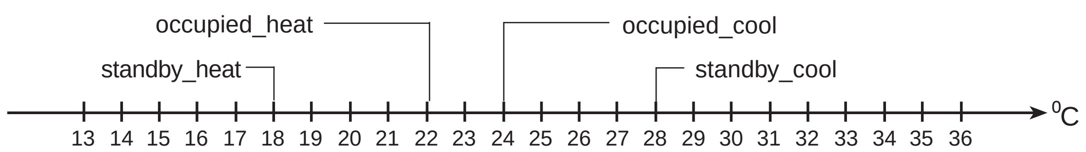
1) Assume that you have the following temperatures in *nciSetpoints*:



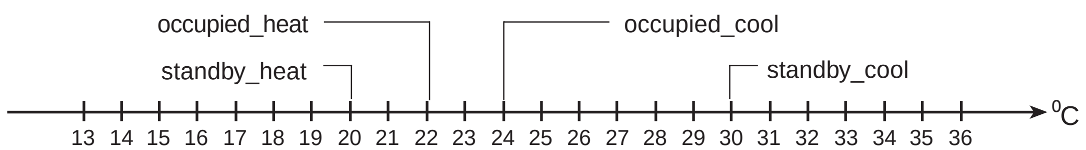
2) You can allow up to 29 °C before you start cooling in economy mode with *nviSetpoint*. The basic setpoint for comfort is 22 °C and for economy 24 °C.



3a) With Method A you get the following result:



3b) With Method B you get the following result:



Appendix B: Commissioning protocol

This protocol can be used when commissioning the Fan Coil controller TAC Xenta 101. Note that the indices are listed in numerical order, not in the order they are used during commissioning. If you need information on accepted values, these are found in the tables in chapter 8.

Index	Function	Variable	Default value	Set value	Note
0	Config. location label	nciLocation	0		
14	Occupancy scheduler input	nviManOccCmd	OC_NUL		
15	Application mode input	nviApplicMode	0=Auto		
16	Zone temperature input	nviSpaceTemp	327,67 °C		
17	Temperature setpoint input	nviSetPoint	327,67 °C		
18	Setpoint offset input	nviSetpntOffset	0 °C		
19	Water temperature input	nviWaterTemp	327,67 °C		
20	Energy hold off input	nviEnergyHoldOff	0=Off, 0 %		
21	Heating control input for slave	nviHeatSlaveR	0		
22	Cooling control input for slave	nviCoolSlaveR	0		
23	Fan control input for slave	nviFanSlave	0=Off, 0 %		
24	Config. application options	nciAppOptions	00000000		
25	Config. occup. temp. setpoints (Cooling setpoint comfort (Cooling setpoint economy (Cooling setpoint off (Heating setpoint comfort (Heating setpoint economy (Heating setpoint off	nciSetpoints occupied_cool standby_cool unoccupied_cool occupied_heat standby_heat unoccupied_heat	 23 °C) 25 °C) 28 °C) 21 °C) 19 °C) 16 °C)		
26*[1]	Config. heat. /cool. changeover	nciXoverLimit	22 °C		
27	Config. max. dev. zone temp.	nciSpaceTempDev	2 °C		
28	Config. min. low limit zone temp.	nciSpaceTempLow	10 °C		
29	Config. max. limit discharge air	nciDischAirMax	35 °C		
30	Config. min. limit discharge air	nciDischAirMin	10 °C		
31	Config. gain for heating contr.	nciGainHeat	25		
32	Config. integral time heat. contr.	nciItimeHeat	900 s		
33	Config. stroke time heat. actuator	nciHeatActStTime	165 s		
34	Config. gain for cooling contr.	nciGainCool	25		
35	Config. integral time cool. contr.	nciItimeCool	900 s		
36	Config. stroke time cool. actuator	nciCoolActStTime	165 s		
37	Config. minimum fan speed	nciMinFanSpeed	0%		
38	Config. zone temp. sensor adj.	nciSpaceTempOfst	0,0 °C		
40	Occupancy sensor input	nviOccSensor	OC_NUL		
41	Config. min. output heat. contr.	nciHeatPrimMin	0%		
42	Config. network conf. source	nciInstallType	0=LOCAL		
43	Config. send heartbeat	nciSndHrtBt	0,0 s		
44	Config. receive heartbeat	nciRcvHrtBt	0,0 s		
45	Object request input	nviRequest	RQ_NUL		
47	File request input	nviFileReq	FR_NUL		

*[1]Only controllers 1VF and 1VFC

Blank page

Index

Z

You can help make this manual even better!

Please help us make our documentation as good as possible. Use this form to let us know of any errors, unclear descriptions or suggested improvements.

Send the form to:

TAC AB
Helpdesk
Jägershillgatan 18
SE-213 75 MALMÖ
SWEDEN

Or e-mail to:

helpdesk@tac.se

I have found the following errors and/or unclear descriptions in the “Xenta 101 Handbook” (part number 0-004-7513-0 (GB)):

On page:...

....

....

On page:...

....

....

I suggest the following improvements:

On page:...

....

....

On page:...

....

....

Name:

Company:

TAC AB, Jägershillgatan 18, SE-213 75 MALMÖ, SWEDEN, +46 40 38 68 50 (switchboard)