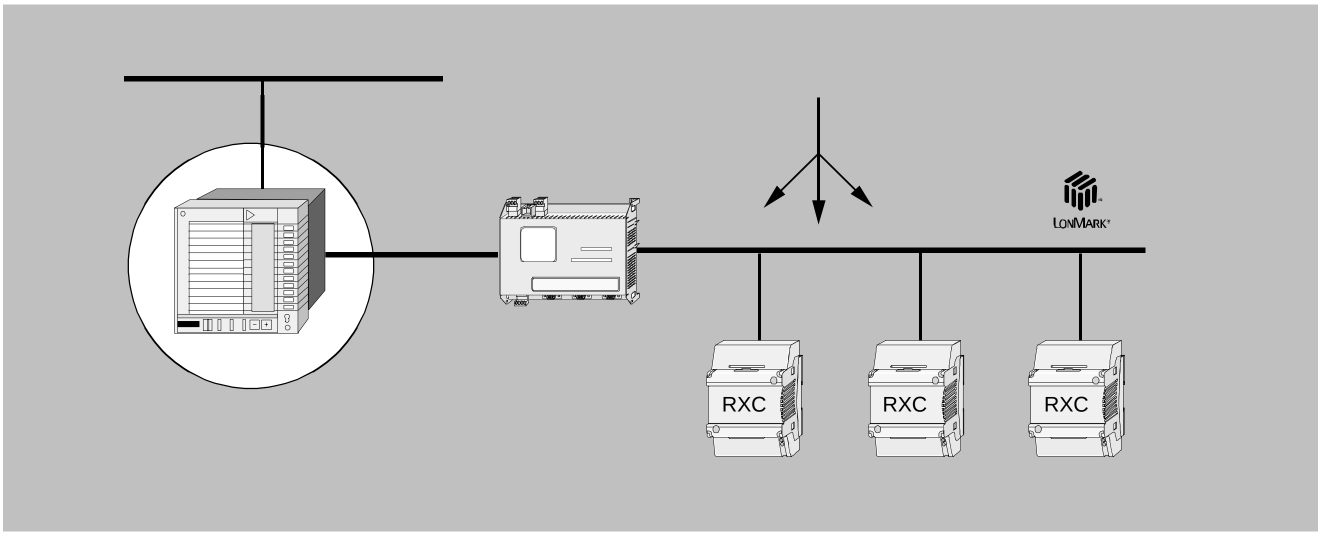
SIEMENS



VISONIK®
Integration of DESIGO™ RXC
Basic documentation

CM2Z8338en_01
12.06.2006

Building Technologies
Building Automation

Contents

1 About this document
1.1 Overview

Introduction

This opening section describes the content and purpose of this document. It

- Introduces the basic topology of the DESIGO RXC system within VISONIK
- Defines the target readership of this manual, and the background knowledge required
- Shows how the activities described fit into the project handling procedure.

Contents of this section

1.2 Revision history

All changes are marked by dots in the right margin.

06.2006

- **Room operation via the Intranet not available any more** **(Section 2.7**
- **New entries to the mapping tables** **(Section 4.10)**

10.2004

- Mapping tables 4.10.1 and 4.10.2: corrections and new entries

01.2004

- Revision of Section 3:
  - 3.2.9 Room setpoint shift
  - 3.2.10 Summer/winter compensation
  - 3.2.21 Room temperature averaging
  - Old subsection 3.6, Intranet room operation, replaced by new subsection 2.7.
  - Old subsections 3.1.33 / 34, Single parameter integration / Third-party inputs/outputs replaced by Tables 4.10.1 and 4.10.2.
- Section 4.10 Single parameter integration, new (Mapping tables 4.10.1 and 4.10.2).

11.2002

- Page 42: Numbering of the DP parameters for switches 2, 3 and 4 for blinds.

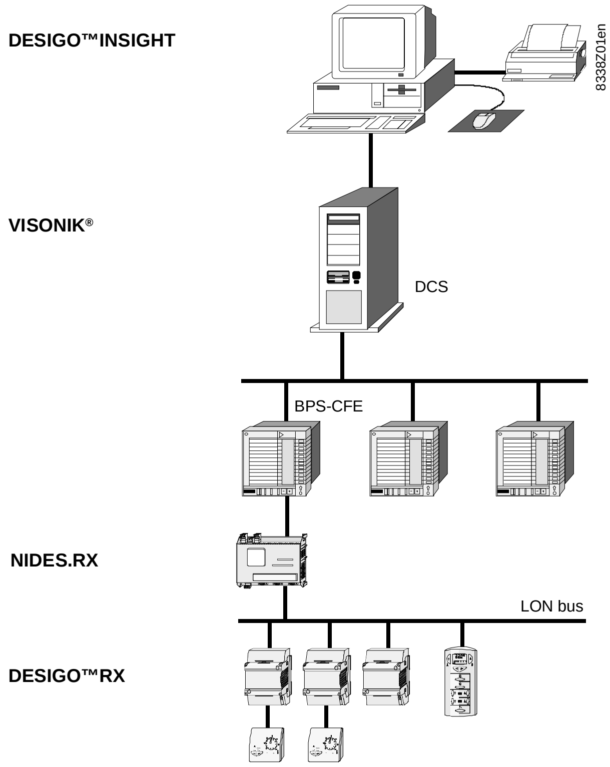
05.2002

- Major changes to Sections 4 and 5.
- Section 6 (SAPIM Code table) and 7 (diagnostics) have been revised and removed to the Engineering Guide, Z8339.

1.3 What is this document about?

DESIGO RXC in VISONIK

This document deals with the integration of the DESIGO RXC room management system into VISONIK. The diagram below shows the basic topology with the associated system levels.



Notes on the diagram

The system levels and components shown in the diagram are:

Name	Description
DESIGO™ INSIGHT	Higher-level management station
VISONIK®	Management and automation level with: – VISONIK DCS – VISONIK BPS Type PRV2.00 (CFE) SW Version ≥ 16.06
NIDES.RX	Interface for the integration of DESIGO RXC, in this case via the VISONIK automation level.
DESIGO™ RXC	Individual room control range based on LON bus/LONMARK technology.

1.4 For whom is this document intended?

Target readership

This document is written for staff in the Building Automation division of Siemens Building Technologies, working in the field of building automation and control as:

- Sales personnel
- System engineers
- Commissioning and service engineers

This document contains all the information relevant to the above staff in relation to the integration of DESIGO RXC into VISONIK.

Background knowledge required

The target readers listed above are assumed to have a basic knowledge of the integrated systems relevant to their responsibilities. The key points are listed below.

DESIGO RXC

The following knowledge of DESIGO RXC is required:

– Range
– Applications
– Integration in general (LON, NIDES)
– RXT commissioning and service tool

VISONIK

In the case of VISONIK, knowledge is required of the following:

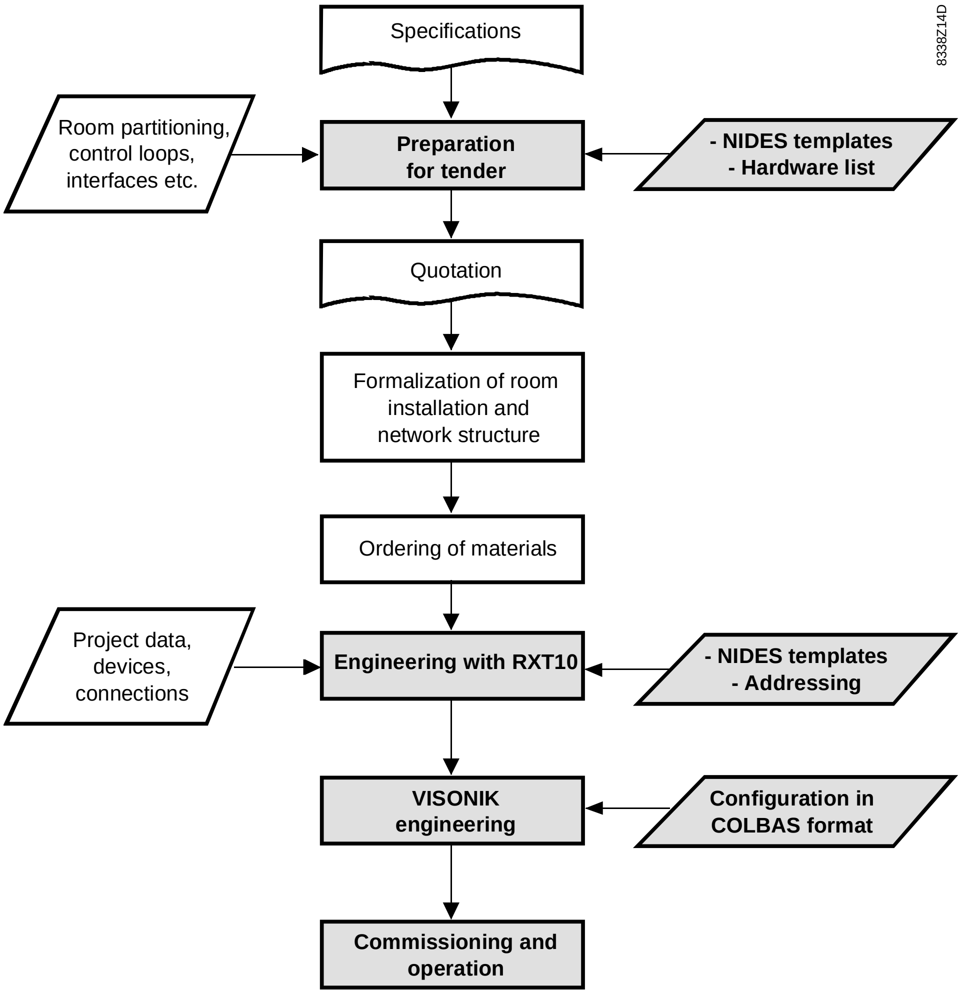
– The overall system
– BPS process image
– Link and FLN points
– The VISOTOOL Editor

Details of the main reference documents for both systems are given at the end of this section under “Related literature”.

1.5 What activities does this document describe?

Project handling from start to finish

The flow chart below is a simplified illustration of the handling of a project for the integration of DESIGO RXC into a management system – from the specification and tender documentation to engineering and commissioning.



Activities described

In the flow chart above, the activities related to VISONIK have been highlighted. The table below provides a brief description of these:

Activity	VISONIK-specific elements
Preparation for tender	Available system functions, NIDES binding templates for VISONIK, hardware list
Engineering with the RXT10 (room-level engineering)	Addressing (location text), NIDES binding templates for VISONIK
VISONIK engineering (Integration into the system level)	Creation of a configuration in COLBAS format.
Commissioning and operation	Loading VISONIK BPS with the COLBAS configuration, completion of the configuration and testing of system functions

These activities, specific to VISONIK, are not described in the related literature listed below. They are the subject of this manual.

1.6 Related literature

Related VISONIK BPS literature

The required information about VISONIK BPS is contained in the documents listed below:

- CM2Z8384e VISONIK COLBAS BPS/DCS Language description
- CM2Z8303e VISONIK Process Image BPS Point Types and Parameters
- CM2Z8568en VISONIK Point Types and Parameter Description VVS20
- CM2T8568en VISONIK System principles VVS20
- CM2B8301e VISONIK BPS User's guide

Related DESIGO RXC literature

The following is a list of the documents containing the necessary information about the DESIGO RXC system:

- DESIGO™ RXC Range description
- DESIGO™ RXC Applications library
- RXT10 commissioning and service tool
- Engineering and commissioning the NIDES.RX LON / LONMARK interface

Engineering guide

A "Quick Guide" for engineering and commissioning is also available to complement this document. This also provides tips on troubleshooting and diagnostics for the CFE link. Document number:

- CA2Z8339en VISONIK Integration DESIGO RXC Engineering Guide

2 Integration overview

2.1 Overview

Introduction

This section gives an overview of the integration of DESIGO RXC into VISONIK. Key items are:

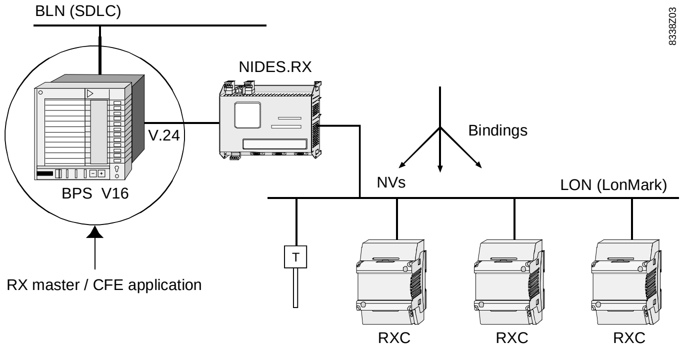
– Integration topology with a brief description of the equipment involved.
– RX master functions and NIDES binding templates
– Preliminary hardware list for an initial estimate of the number of controllers

Contents of this section

2.2 Topology for RX integration

Structure

The diagram below shows the topology for the integration of DESIGO RX into VISONIK.



Components involved

The main components involved in the integration of RX into VISONIK are:

Element	Description
BLN (SDLC)	Building Level Network (SDLC ring) of the VISONIK system
BPS V16	VISONIK BPS, type PRV2.00, from Version 16, with large memory board type PVA3.02 and the CFE “RX master” application for the integration of DESIGO RX into VISONIK.
V.24	The VISONIK BPS is connected to the NIDES via one of the two V24 ports, TTY1 and TTY2. (Communication with the TRUNKBUS protocol)
NIDES.RX	Interface via which the DESIGO RXC devices are integrated into the automation level. The NIDES also acts as a LON ⇔ V.24 gateway. Each BPS can accommodate one NIDES with max. 900 RS data points.
RXC	DESIGO RXC room controllers The controller functions are determined by downloadable application software, referred to as the “application”.
LON (LONMARK)	**L**ocal **O**perating **N**etwork. Connects the DESIGO RXC controllers with each other and with other LONMARK-compatible equipment.
NVs	The devices on the LON bus communicate by means of network variables (NVs). These units of information contain the data in a structured format based on the LONMARK standard.
Bindings	The “bindings” are defined at the project engineering stage and downloaded with the application in the commissioning phase. They identify the network variables to be exchanged between individual LON devices. Examples: – In the diagram above, the changeover thermostat (T), a LONMARK-compatible device, transmits the changeover command to all the defined DESIGO RXC controllers. – All DESIGO RXC controllers transmit changes of value to the NIDES, in accordance with the NIDES binding template.

2.3 Mapping the RXC controllers in the RX master

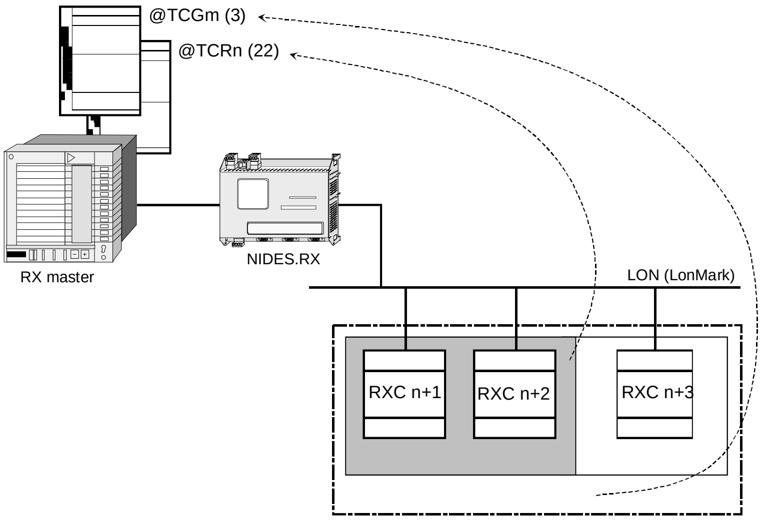
Mapping as VISONIK data points

As the RX master, the VISONIK BPS maps the DESIGO RXC application to VISONIK data points and parameters as follows:

- **TEC data points** for HVAC functions (TEC groups and TEC rooms)
- **Link points** for third-party blinds, lighting and special functions

Functions

The mapped data points can be operated like normal VISONIK points. As an example, the diagram below shows operation of a TEC group and TEC room at the BPS by use of POP cards:



Notes on the diagram

The table below provides further information on the elements in the diagram above:

Element	Description
RX n+1, etc.	All LON devices are identified by a unique address before they leave the factory (the neuron address). This is used to identify the devices on the LON. At the engineering stage, however, each RXC device is assigned a VISONIK address with a controller number (RXC n+1 etc.) and a room number.
NIDES.RX	The NIDES.RX converts the data received from the DESIGO RXC controllers into RS data points, and passes these to the RX master. (RS data points are the units of information used in the AS1000/MS2000 systems of the Building Automation division of SBT.)
RX master	The VISONIK BPS as RXC master: – Maps the RS data points from the NIDES to virtual TEC structures and link points, and vice versa – Makes available a range of functions – the RX master functions –for coordination, control and operation
@TCGm (3) @TCRn (22)	At the BPS, access is not via the network-based LON address of the RXC controllers, but via the application-based room and group addresses assigned in the engineering phase, as shown in the example above for: – TEC group 3 (@TCG3) – TEC Room 22 (@TCR22) *Note:* Access to individual controllers takes place indirectly, via the room, as usual for TEC controllers.

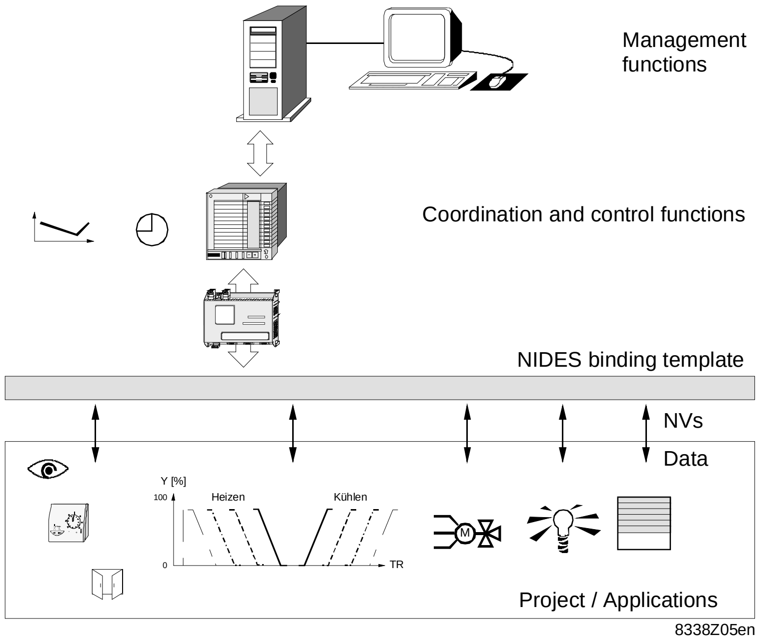
2.4 RX master functions / NIDES binding templates

The concept of RX master functions

The actual purpose of integration is to make available the RX master functions . These either incorporate the following higher-level functions directly, or they render them possible by indirect means:

- Management functions at the management level (DESIGO INSIGHT or VISONIK Insight)
- Coordination and control functions at the automation level (VISONIK BPS)

RX master functions include options such as operating modes, time-switch catalogs, summer/winter compensation, setpoint adjustments etc., as illustrated below.



The NIDES binding templates

To implement the RX master functions, the BPS needs current data from the applications. This data is available in the network variables (NV) on the LON bus. Defining the relevant NIDES bindings provides the BPS with access to the required data. The RXT10 commissioning and service tool includes the following facilities for this purpose:

– Use of the pre-defined VISONIK Templates 1 and 2
– Creation of new templates by adding or omitting network variables from these templates.

Caution: Do confuse the NIDES bindings with those used internally and between the LON devices. They each have their own templates in the RXT10 tool.

Scope of RX master functions

The scope of the RX master functions depends on the following:

– The functional scope required by the client at the management and automation levels
– The maximum allowable number of integrated controllers per NIDES / BPS

It is often necessary to reach a compromise between these two sets of circumstances. The next topics therefore provide information about the RX master functions with the VISONIK Templates 1 and 2, and about the additional functions available. This is followed by a preliminary hardware list for a preliminary estimate.

2.5 RX master functions with VISONIK templates 1 / 2

Introduction

VISONIK Templates 1 and 2 are stored in the RXT10 as default templates for integration into the NIDES.RX. The difference between the two templates is as follows:

– VISONIK Template 1 contains the main functions for HVAC, blinds and lighting
– VISONIK Template 2 incorporates additional HVAC functions

These functions are described in detail in the two tables below.

HVAC functions

The following shows the RX master functions for HVAC applications, available when using VISONIK Templates 1 and 2. The additional functions offered by Template 2 are shown on a shaded background.

Template 1	Template 2	RX master function
✓	✓	Definition of the HVAC operating modes: Comfort, Pre-comfort, Economy, Standby
✓	✓	Display of current operating mode and associated current setpoints
	✓	Read/Write cooling and heating setpoints
✓	✓	Summer/winter compensation and room setpoint shift
✓	✓	Display current room temperature
	✓	Display of valve positions
	✓	Display current volumetric air flow
	✓	Display current fan speed
✓	✓	Display of controller status: Normal / Fault
✓	✓	Room temperature monitoring (TEC function, needs to be configured)
✓	✓	Optimum Start Program (TEC function, needs to be configured)
✓	✓	Summer/winter changeover
✓	✓	Define controller operating mode: Heating, Cooling, Night purge etc.

Blinds and lighting

The next table shows the basic functions for blinds and lighting. Both Template 1 and Template 2 incorporate these functions.

Template 1	Template 2	RX master functions
✓	✓	Display of effective blind status of blinds 1 to 4
✓	✓	“Local” / “Remote” command for blinds 1 to 4
✓	✓	Display of effective lighting status of blinds 1 to 4
✓	✓	“Local”/“Remote” command for lighting zones 1 to 4

2.6 RS master functions, extended selection

Introduction

The RX master functions automatically integrated with VISONIK Templates 1 and 2 can be supplemented with a number of other functions if required. These are listed below.

HVAC functions

The following additional HVAC functions are available:

- Display of current heating or cooling demand
- Display of active heating or cooling sequence
- Display window contact
- Display current damper position
- Define emergency operation:
  Emergency OFF / Smoke extraction
- Room temperature averaging
- Disable or limit auxiliary energy supply

Blinds and lighting

The following additional functions are available for lighting and blinds:

- Write individual blind states in “Local” operating mode:
  Up, Down 1, Down 2
- Write individual lighting-zone states in “Local” operating mode:
  OFF (0), ON (1..100%)

Single-parameter integration

For all the RX master HVAC functions described above – whether available as standard in the VISONIK Templates 1 and 2, or from the extended selection – the associated network variables are mapped as TEC data points in the RX master.

Each RXC application contains a number of additional network variables, which can also be integrated. This is achieved by means of single-parameter integration via link points. Examples of the application of single-parameter integration are:

Application example	**Details**
Representation of individual RXC controller I/Os RXC31.1 / RXC32.1	For the integration of applications where the RXC controllers are used as universal I/O modules, e.g. to: – Interrogate switch states – Control field devices on the basis of system functions See Applications library CA2A3813E: Basic applications / 000 applications The free I/Os of these two room controllers are described in the VAV function description CA2A3817.
Third-party devices	Integration of LONMARK-compatible third-party devices

2.7 Room operation via the Intranet

Phased out

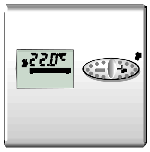
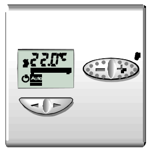
From 2006, the room operation via the intranet is not available any more.

Introduction

Room operation via the Intranet is designed for office buildings with PC networks installed. In these circumstances, instead of wall-mounted room units, users can operate the room from their office PC via an Internet browser.

Room units and functions

The diagram below shows the room units in the DESIGO RXC range, and their functions.

Function	**Type 1** 	**Type 2** 
Display current room temperature	✓	✓
Read/write room operating mode (Standby / Auto)	–-	✓
Read/write room unit setpoint adjustment	✓	✓

Binding Template

A separate template is available for integration of data points for Intranet room operation.

Notes

Note the following in relation to Intranet room operation:

– By definition, only the room unit for the master controller in the room is operated.
– The setpoint adjustment is limited to the range –3K to +3K.

2.8 Preliminary hardware list

Overview

The table below shows a preliminary hardware list. It provides the following information for the various applications, and can therefore be used for a preliminary estimate when preparing a quotation.

- Possible number of RXC controllers per NIDES
- Number of VISONIK points in the BPS

Application	Number of DESIGO RX controllers per NIDES		Approx. number of VISONIK points in RX master
	VISONIK Template 1	VISONIK Template 2	
Radiator-type systems	80	47 … 50	140 … 240
Radiator / chilled ceilings	80	47 … 50	135 … 240
Fan-coil units	80	42 … 47	120 … 240
Integrated application (HVAC, blinds, lighting)	42 … 60	30 … 39	220 … 420
VAV	80	42 ... 50	120 ... 240

Notes on the overview

Note the following in relation to the information in the table:

– Where the number of RXC controllers is given as a range, this is because individual applications in the category concerned have a different number of network variables (as for example in VISONIK Template 2 for applications RAD01 and RAD03 in the “Radiators” category).
– The estimate of the number of controllers per NIDES is based on a maximum of 900 RS data points in the NIDES. However, it is advisable to plan a reserve capacity of 10% . (in other words, to use no more than 800 data points, if possible) to accommodate future changes and enhancements etc.
– The term “VISONIK points” refers to TEC points and link points. These must not exceed 500 in number.

Detailed calculation

For a guide to detailed calculation refer to Section 4, "Engineering".
This will help you determine more precise quantities, taking the following into account:

– Different applications (mixed)
– Additional RS master functions to be integrated from the extended selection.

3 Integration functions

3.1 Overview

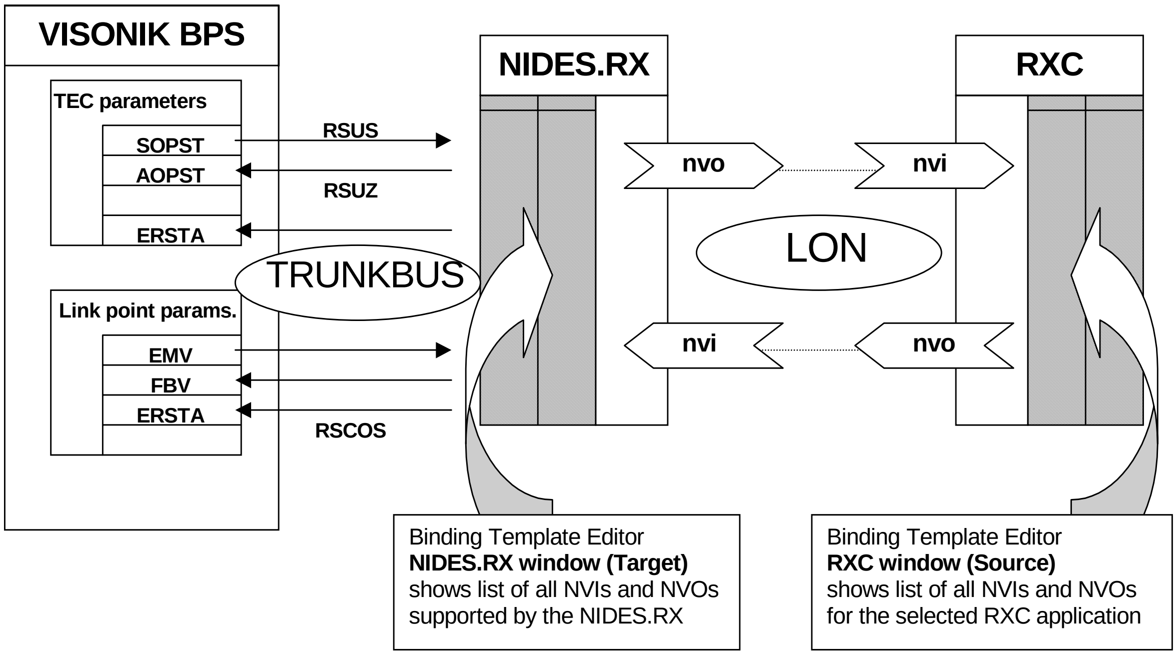
Introduction

In the first part of this section, all the RX master functions available for the RXC applications are described in detail. Subsections 3.3 to 3.6 deal with the creation of groups for HVAC, lighting and blinds.

Contents of this section

3.2 Functions for the DESIGO™ RXC applications

The RX master functions in the BPS are implemented by mapping VISONIK parameters to controller network variables. As can be seen in subsection 2.2, the VISONIK parameters are first transmitted to the NIDES.RX and only then mapped to the network variables. The following diagram shows the process from the functional viewpoint.



The parameter values in are exchanged as RS data points in the trunkbus protocol via the V24 link between the BPS and NIDES.RX[1]

Network variables are exchanged on the LON side. This involves the connection of controller inputs (nvi) to NIDES.RX outputs (nvo) and vice versa.

These connections, or “bindings” can be drawn in the Binding Template Editor of the RXT10 tool. They are already defined in the standard VISONIK Binding Templates.

For the HVAC functions, the parameters used are primarily those of the TCR and TCG points. For lighting and blinds, the link point parameters EMV (Event Main Value), FBV (Feedback Value) and ERSTA (Error State) are used.

For each RX master function, the mapping of the relevant parameters to the associated network variables is described in the subsections which follow. The first subsection explains the symbols used, with “Intranet room operation” as an example.

Note for third-party devices

With integration into VISONIK, it is also possible to integrate network variables for third-party devices. However, the VISONIK system will not recognize the associated applications. The general procedure for the integration of all kinds of network variables is described in Section 4, Engineering, subsection 4.11, Integration of single parameters.

[1] The “Trunkbus protocol” and the “RS data points” are concepts associated with the MS200 system (see also 4.2). Each RS data point represents a single NV element. However, the user does not need to be concerned with this.

Definition of RS master function

NV name in NIDES.RX

NV name in RXC controller

Shows which RXC applications can be used

3.2.1 Example of a read operation

3.2.24 Intranet room operation

All RXC applications

VISONIK BPS			NIDES.RX			RXC controller			
DP. Parameters	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.AOPST	See "Effective room status"	←	nviEffectOccup	See "Effective room status"	←	1	Effect Occup	nvoEffectOccup	
@TCRy.ARTP	Current room temperature	←	nviSpaceTemp	Current room temperature	←	1	Effect Space Temp	nvoSpaceTemp	
@AOx'24.EMV	Room status: 0 = Occupied 3 = Standby	→	nvoOccManCmd	Operating mode on room unit	→	ne	Occup Override	nviOccManCmd	
@AOx'12.EMV	Setpoint adjustment : -3 ... +3 °C	→	nvoSetptOffset	Setpoint adjustment from room unit	→	ne	Setpt Offset	nviSetptOffset	
@AI x' 80.EMV	Actual setpoint adjustment	←	nviSetptOffset	Actual setpoint adjustment	←	ne	Local Setpt Offset	nvoSetptOffset	

TEC Room parameter
n = Room No. (1..99)

Direction of data flow

Link point AI / AO, Address x' y ,
x = Controller No. (33..126)
y = Code for function or NV
y = 1..4 : Blind function (AO)
y = 5..8 : Lighting function (AO)
y =11..210 : Single parameter of NV (AI / AO)

Standard Binding Template, containing the relevant network variable:
1 / 2 / ne = Not in any standard binding template

Former designation for network variablein the Binding Template Editor of the RXT10 tool. Now only found in the documentation.

3.2.2 Displaying the device status

All RXC applications and all third-party devices

VISONIK BPS			NIDES.RX				RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*		*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.ERSTA	Status display for whole room: 0 = All controllers normal, 9 = At least one faulty controller (no communications)	←	nviStatus		LON device status normal / faulty	←	1	Object Status	nvoStatus	
@AI x' y.ERSTA @AO x' y.ERSTA	Status display, single value: 0 = Controller x normal, 9 = Controller x fault									

Caution The network variable Object Status must be integrated in each device. This is especially important for third-party devices!
Without this variable, parameter ERSTA remains at a value of 9 (disconnected) in respect of the room point and all link points to the device.

Note The cause of error @TCRn.ERSTA=9 may either be a fault in the controller or a communications problem between the controller and the NIDES.RX. If an error occurs, it takes some 5 minutes to be registered by the VISONIK BPS. When the room controller resumes normal operation, the message is registered by VISONIK BPS with minimal delay.

3.2.3 Definition of room operating mode

All RXC applications

VISONIK BPS	
DP.Parameter	*Function/Range*
@TCRn.SOPST	See table of bindings
@TCRn.MOPST [1]	
@TCRn.SOMOD	

NIDES.RX	
NV name / element	*Function/Range*
nvoUseSchedule	See table of bindings
nvoOccSchedule	See table of bindings

	RXC controller		
BT	*User designation*	*NV name*	NV element
1	Building Use Scheduler	nviUseSchedule	Current state
1	Occup Scheduler	nviOccSchedule	Current state

[1] Manual Operating State: Same function as @TCRn.SOPST when @TCRn.MACT=1

Table of bindings:

RX master (BPS)		Network variables		RXC controller	RX operator unit
@TCRn .SOMOD	**@TCRn .SOPST**	**Building Use Scheduler**	**Occup Scheduler**	**RXC... Operating mode**	**Operation on QAX... ⏻ ↔ Auto**
0 = Local	**0 = Comfort**	**Occupied**	**Occupied**	**Comfort**	**Standby ↔ Comfort**
	1 = Reduced	**Occupied**	**Standby**	**Standby**	**Standby ↔ Comfort**
	2 = Economy	**Occupied**	**Unoccupied**	**Economy**	**Economy ↔ Comfort***
	3 = Standby	**Unoccupied**	**Unoccupied**	**Bldg protection**	**Not possible**
= Remote !	**0 = Comfort**	**Occupied**	**Occupied**	**Comfort**	**Standby ↔ Comfort**
!	**1 = Reduced**	**Occupied**	**Standby**	**Standby**	**Standby ↔ Comfort**
	2 = Economy	**Standby**	**Unoccupied**	**Economy**	**Not possible**
	3 = Standby	**Unoccupied**	**Unoccupied**	**Bldg protection**	**Not possible**

! Functions not same as with TEC

* Override function (timer acti

Initial state The CFE software @TCRn.SOPST=0 (Comfort) is set after a restart following the loading of a new project.

Note From CFE Version 2.1, the operating mode is only written to the controllers after a restart if the associated option has been set. For more on this subject, refer to the V2.1 release notes.

3.2.4 Define controller operating mode

All RXC applications

VISONIK BPS			NIDES.RX				RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.RLHT @TCRn.RLCO	See table of bindings	→	nvoApplicMode	Controller application mode (see table of bindings)	→	**1**	Application Mode	nviApplicMode	

Table of bindings

@TCRn.RLHT	@TCRn.RLCO		Application Mode	Description
1	1	⇨	0 = HVAC_AUTO	Normal operation (heating and cooling possible)
1	0	⇨	1 = HVAC_HEAT	Heating only (cooling sequence disabled)
0	1	⇨	3 = HVAC_COOL	Cooling only (heating sequence disabled)
0	0	⇨	9 = HVAC_FAN_ONLY	Ventilation only (heating and cooling sequence disabled)

In changeover applications, therefore, the changeover has to be effected between the two operating modes HVAC_HEAT (winter) and HVAC_COOL (summer) As can be seen from the binding table, new values must be assigned in each case to the two parameters RLHT and RLCO. For this purpose it is recommended that you should effect the changeover via the intermediate state RLHT / RLCO = 1 / 1 (i.e. HVAC_AUTO).

Note The question of which operating modes are actually supported depends, of course, on the application in the RXC controller. For radiator applications, for example, the operating modes HVAC_COOL and HVAC_FAN_ONLY have no effect.

Caution The NV “Application Mode” is also used by the “Night purge” function. Since the last-received value is always valid in the RXC controller, the functions can influence each other mutually.

Note The other values of the NV "Application Mode" are only supported via the associated link point from CFE Version 2.1.

3.2.5 Night purge

All VAV applications, FPB05

VISONIK BPS			NIDES.RX				RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.SOPST	See table of bindings	→	nvoUseSchedule	See table of bindings	→	1	Building Use Scheduler	nviUseSchedule	Current state
@TCRn..RRTP		→	nvoOccSchedule	See table of bindings	→	1	Occup Scheduler	nviOccSchedule	Current state
@TCRn.SPRH		→	nvoApplicMode	See table of bindings	→	**1**	Application Mode	nviApplicMode	

Table of bindings

@TCRn.SOPST=4	**@TCRn.RRTP > (@TCRn.SPRH-1)** **SOPST remains at 4**	**@TCRn.RRTP ≤ (@TCRn.SPRH-1)** SOPST goes to 3 (Standby)
Building Use Scheduler	OCC_OCCUPIED	OCC_UNOCCUPIED
Occup Scheduler	OCC_UNOCCUPIED	OCC_UNOCCUPIED
Application Mode	HVAC_NIGHT_PURGE	HVAC_AUTO

The night purge function is activated by the operating state command @TCRn.SOPST=4. The mode is canceled when the current room temperature @TCRn.RRTP falls to 1K below the reduced heating setpoint @TCRn.SPRH.

Caution If the room is subject to a time scheduler @DSTy , and if the Night Purge function is not activated by this function, it is important to ensure that the parameter @DSTy.MODE is set to 0, as otherwise, the parameter @TCRn.SOPST will immediately be overwritten with another value.

Caution The NV “Application Mode” is also used to specify the controller operating mode (e.g. changeover). Since the last-received value is always valid in the RXC controller, the functions can influence each other mutually.

3.2.6 Displaying the actual controller operating mode

All RXC applications

VISONIK BPS			NIDES.RX				RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*		*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.AOPST	See table of bindings	←	nviEffectOccup [2)]		See table of bindings	←	**1**	Effect Occup	nvoEffectOccup [2)]	

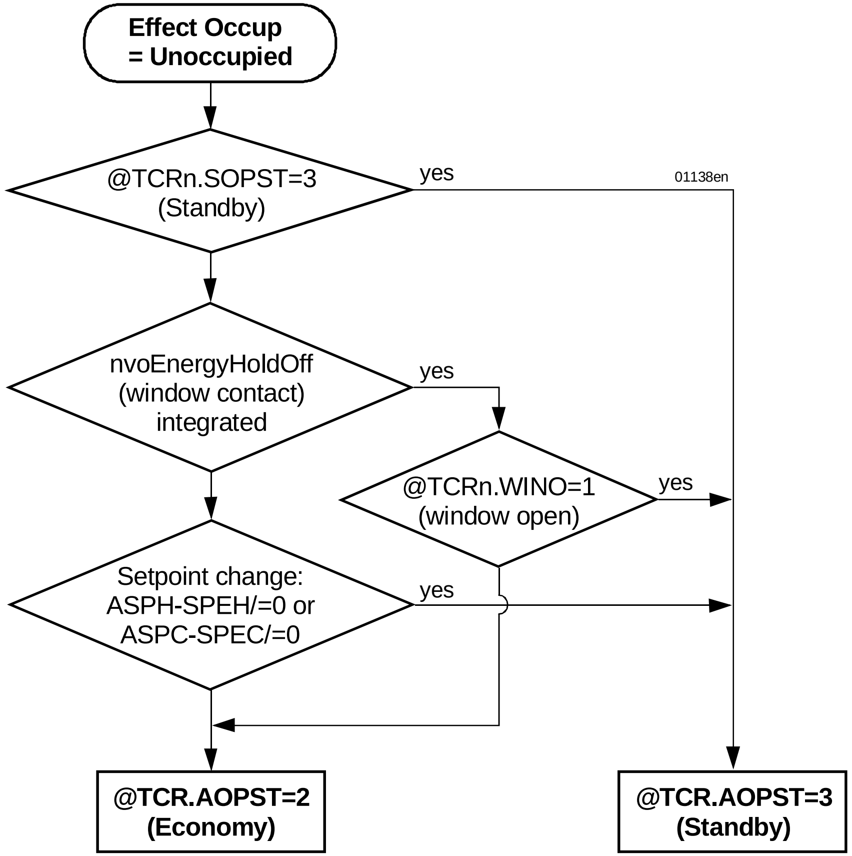
[2)] In the RXC Application Library V2, the NV element EffHvacMode.op_mode is used instead of NV EffectOccup.

Table of bindings

Effect Occup	**@TCRn.AOPST**
0 = Occupied	0 = Comfort
3 = Standby	1 = Pre-comfort
1 = Unoccupied [1)]	2 = Economy
	3 = Standby

[1)] The NV “Effect Occup” does not distinguish between the TEC operating states Economy and Standby.
The effective operating state is therefore determined by the CFE-SW in accordance with the following flow chart.
In order for this to work properly, either the window contact (nvoEnergyHoldOff) or the basic setpoints (nciSetpoints) must be integrated (see further below).
If neither of these is integrated, then ensure that the basic setpoints, set in the controllers with the RXT tool match those in the BPS.

Note: From CFE Version 2.1 on, there is no longer a "Setpoint change" test. Refer to the V2.1 release notes (Intranet).



3.2.7 Display current room temperature

All RXC applications

VISONIK BPS			NIDES.RX				RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.ARTP(x)	Current room temperature Controller x	←	nviSpaceTemp	°C	←	**1**	Effect Space Temp	nvoSpaceTemp	

3.2.8 Display effective setpoints for heating and cooling

All RXC applications

VISONIK BPS			NIDES.RX				RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.ASPH	Effective heating setpoint	←	nviSetptEffect .occupied_heat [2)]	°C	←	**1**	All Effect Setpts	nvoSetptEffect [2)]	occupied_heat
@TCRn.ASPC	Effective cooling setpoint	←	nviSetptEffect .occupied_cool [2)]	°C	←	**1**	All Effect Setpts	nvoSetptEffect [2)]	occupied_cool

[2)] In the RXC Application Library V2, the NV elements EffHvacMode.cool_setpt and heat_setpt are used instead of the NV SetPtEffect.

3.2.9 Room setpoint shift

All RXC applications

VISONIK BPS			NIDES.RX				RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.RRSC	Room setpoint shift range – 5...+5°C	→	nvoSetptShift .occupied_cool	Occupied cooling setpoint shift	→	**1**	Setpt Shift	nviSetptShift	occupied_cool
		→	nvoSetptShift .occupied_heat	Occupied heating setpoint shift	→	**1**	Setpt Shift	nviSetptShift	occupied_heat

Caution The room setpoint shift to for an HVAC group is effected via a VIP point. For each @TCGm HVAC group the CFE software automatically generates an associated @VIPm. Values written to this point are passed on to the @TCRn.RRSC parameters of all group members.

Note The value x of RRSC is mapped as follows:
x > 0 : occupied_heat = x, occupied_cool = 0
x < 0 : occupied_heat = x, occupied_cool = x
In both cases, the setpoints in the controller are shifted in parallel in this process.

3.2.10 Summer / winter compensation

All RXC applications

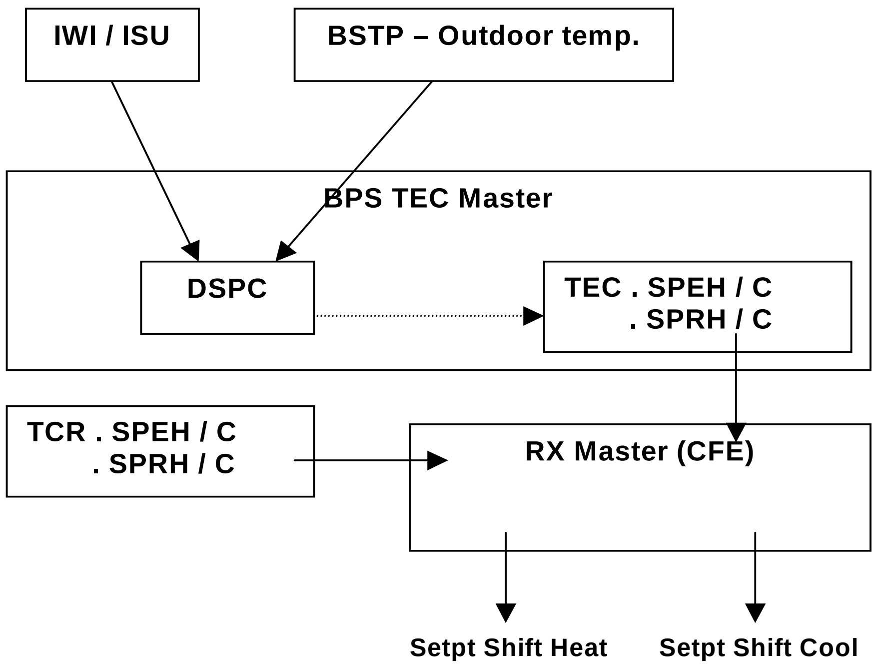
VISONIK BPS			NIDES.RX			RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCGm.DSPC	See diagram	→	nvoSetptShift .occupied_cool	Occupied cooling setpoint shift	→	**1**	Setpt Shift	nviSetptShift	occupied_cool
@TCGm.ISU / IWI		→	nvoSetptShift .occupied_heat	Occupied heating setpoint shift	→	**1**	Setpt Shift	nviSetptShift	occupied_heat

Summer/winter compensation is a function in the group point (@TCG...). The resulting adjustment value @TCGm.DSPC is written to the cooling/heating setpoint shift as a function of the parameters TCGm.ISU (=summer influence) and @TCGm.IWI (=winter influence).

Note The correction value DSPC is calculated in the system software of the VISONIK BPS (TEC Master) from the differential represented by the base temperature @TCGm.BSTP minus the outdoor temperature at address @TCGm.SIDTO. The function is only enabled if SIDTO is pointing to a valid value (e.g. SIDTO=#VIP1), and if only one of the parameters IWI / ISU is a value other than zero. The function shifts the setpoints SPCH / SPCC in the TEC points (see drawing) in accordance with the value of DSPC . However, the room setpoints @TCRm.SPCH and SPCC remain unchanged.

Caution The two functions “Room setpoint shift” and “Summer/winter compensation” act on the same network variables and can, therefore, influence each other.

Note **A**n adjustment x of the heating setpoint @TEC.SPCH also causes the the value of SetptShift occupied_cool to be written
x > 0 : occupied_cool = - x
x < 0 : occupied_cool = 0
This ensures that the effective cooling setpoint in the controller does not change.
An adjustment x of the cooling setpoint @TEC.SPCC only results in the value of SetptShift occupied_cool being written. The effective heating setpoint in the controller is not changed.



3.2.11 Basic setpoints

All RXC applications

VISONIK BPS			NIDES.RX					RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.SPCH	Comfort heating setpoint	↔	ncoSetpoints . occupied_heat	Occupied heating setpoint	↔	**2**	Occup Setpts	nciSetpoints	occupied_heat
@TCRn.SPRH	Reduced heating setpoint	↔	ncoSetpoints . standby_heat	Standby heating setpoint	↔	**2**	Occup Setpts	nciSetpoints	standby_heat
@TCRn.SPEH	*Economy* heating setpoint	↔	ncoSetpoints . unoccupied_heat	Unoccupied heating setpoint	↔	**2**	Occup Setpts	nciSetpoints	unoccupied_heat
@TCRn.SPCC	Comfort cooling setpoint	↔	ncoSetpoints . occupied_cool	Occupied cooling setpoint	↔	**2**	Occup Setpts	nciSetpoints	occupied_cool
@TCRn.SPRC	Reduced cooling setpoint	↔	ncoSetpoints . standby_cool	Standby cooling setpoint	↔	**2**	Occup Setpts	nciSetpoints	standby_cool
@TCRn.SPEC	Economy cooling setpoint	↔	ncoSetpoints . unoccupied_cool	Unoccupied cooling setpoint	↔	**2**	Occup Setpts	nciSetpoints	unoccupied_cool

Caution The basic setpoints are stored in a non-volatile memory in the RXC controller with a limited number of write cycles. It is therefore extremely important to avoid frequent modification of any of these values! Instead of using a time scheduler, for example, to modify the setpoints on the basis of the time of day, the same effect can be achieved by changes the operating mode (e.g. between *Comfort* and *Pre-comfort.* It is also possible to implement temporary setpoint changes via a setpoint shift (@TCRn.RRSC).

Note From CFE Version V2.1 the basic setpoints can, optionally, be synchronized when the controller is powered on.
Refer to the V2.1 release notes on this subject.

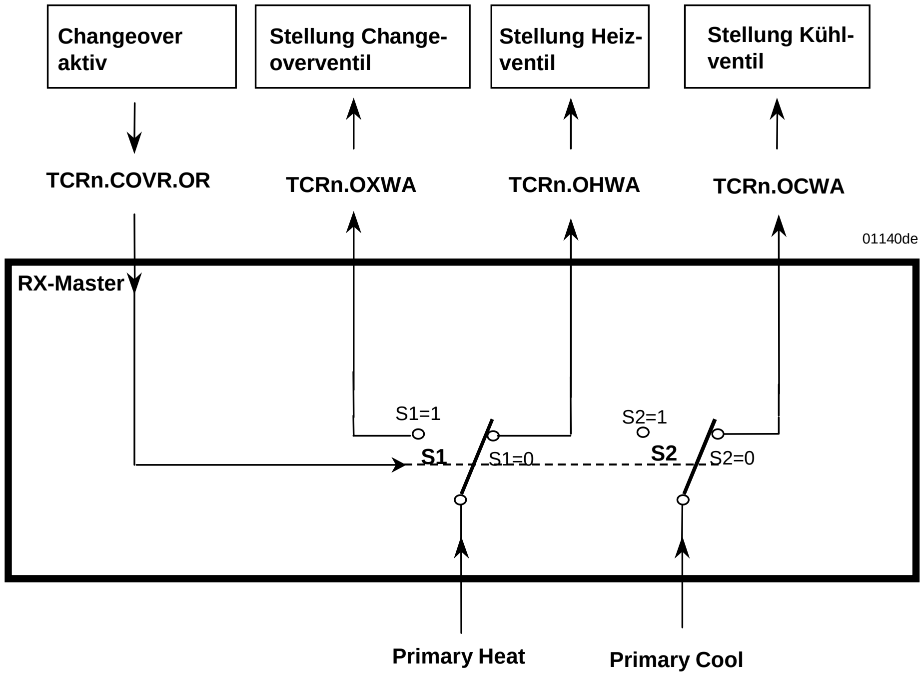
3.2.12 Display effective valve positions

All RXC applications

VISONIK BPS			NIDES.RX					RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.COVR.OR @TCRn.OHWA @TCRn.OXWA	Position of heating or changeover valve (see diagram)	←	nviHeatPrimary	Position of primary heating valve (0…100%)	←	**2**	Primary Heat	nvoHeatPrimary	
@TCRn.OCWA	Position of cooling valve (see diagram)	←	nviCoolPrimary	Position of primary cooling valve (0…100%)	←	**2**	Primary Cool	nvoCoolPrimary	

Note Depending on the type of application, it may be that only “Primary Heat” or only “Primary Cool” is incorporated.

Caution With changeover applications, the current position of the control valve of the RXC controller is always sent to nvoHeatPrimary. As it is not possible to distinguish between heating and cooling without additional information (integration of the NV “Unit Status”), the current valve position is mapped to @TCRn.OXWA.



3.2.13 Display current fan speed

All FNC applications

VISONIK BPS			NIDES.RX			RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.OFAN	Fan speed	←	nviFanSpeed	0 = Off, 1 = Sp.1, 2 = Sp.2, 3 = Sp.3	←	**2**	Fan Speed	nvoFanSpeed

3.2.14 Electric heating coil, position of current valve and max. electrical energy

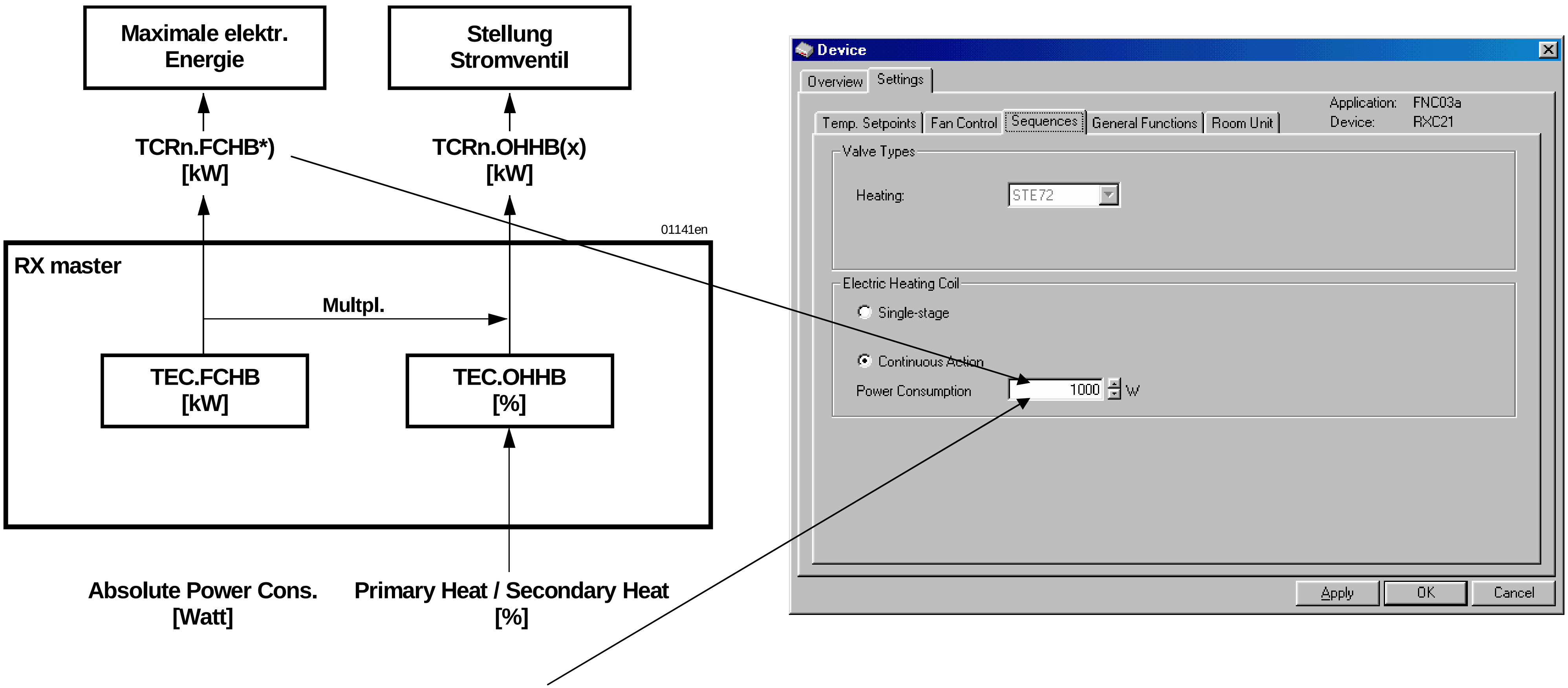
RAD03, FNC01, VAV03, VAV06, FPB05

VISONIK BPS			NIDES.RX			RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.FCHB	Max. electrical energy (see diagram)		nviLoadAbs	Heating output in watts	←	**2**	Absolute Power Cons.	nvoLoadAbs
@TCRn.OHHB(x)	Position of current valve, controller x	←	nviHeatPrimary	Position of primary heating valve	←	**2**	Primary Heat	nvoHeatPrimary

3.2.15 Electric reheater, position of current valve and max. electrical energy

FNC03

VISONIK BPS			NIDES.RX			RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.FCHB	Max. electrical energy (see diagram)		nviLoadAbs	Heating output in watts	←	**2**	Absolute Power Cons.	nvoLoadAbs
@TCRn.OHHB(x)	Position of current valve, controller x	←	nviHeatSecondary	Position of secondary heating valve	←	**2**	Secondary Heat	nvoHeatSecondary



The parameter FCHB must be configured in accordance with the maximum output of the electric reheater.
The value must be the same as the one configured in the RT10 tool (see diagram, right).

3.2.16 Display of current air flow rates

VAV applications, FPB05

VISONIK BPS			NIDES.RX			RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.AIRVS2(x)	Supply air flow rate (cooling duct)	←	nviAirflow	Volume flow in l/s	←	2	Airflow	nvoAirflow
@TCRn.AIRVS1(x)	Supply air flow rate, heating duct	←	nviAirflowHeat	Volume flow in l/s	←	2	Airflow Heat	nvoAirflowHeat
@TCRn.AIRVE(x)	Extract air flow volume	←	nviAirflow2	Volume flow in l/s	←	2	Airflow 2	nvoAirflow2

Note Depending on the type of application, it may incorporate only "Airflow", or "Airflow and "Airflow 2".

3.2.17 Display current damper positions

VAV applications, FPB05

VISONIK BPS			NIDES.RX			RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.OXAIR(x)	Supply air damper position (cooling duct)	←	nviAirDamperPos	Damper position 0..100 %	←	ne	Air Damper Pos	nvoAirDamperPos
@TCRn.OHAIR(x)	Supply air damper position (heating duct)	←	nviHeatAirDmpPos	Damper position 0..100 %	←	ne	Heat Air Damper Pos	nvoHeatAirDmpPos
@TCRn.OEAIR(x)	Extract air damper position	←	nviAirDamperPos2	Damper position 0..100 %	←	ne	Air Damper Pos 2	nvoAirDamperPos2

Note Depending on the type of application, it may be that only "Air Damper Pos" or "Air Damper Pos" and "Air Damper Pos 2" are incorporated.

3.2.18 Display window contact

All RXC applications

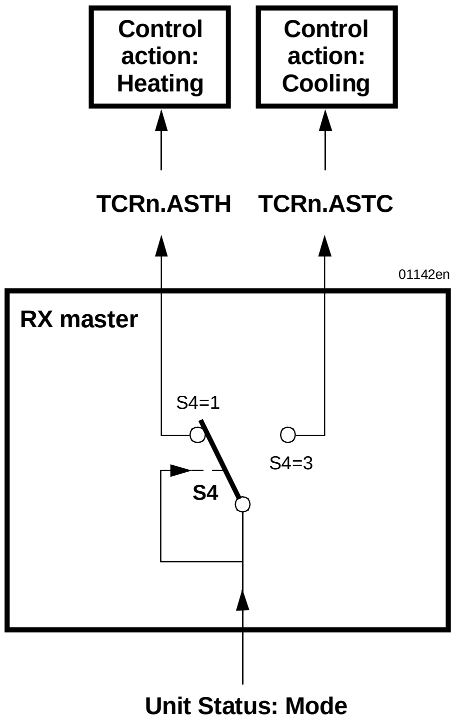
VISONIK BPS			NIDES.RX			RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.WINO	Window contact	←	nviEnergyHoldOff .state	0 = Normal, 1 = Energy holdoff	←	**ne**	Energy Hold-off	nvoEnergyHoldOff	State

Note In order to display the window contact correctly in *Standby* mode, it must be integrated at room controller level.

3.2.19 Display current sequence (heating / cooling)

All RXC applications

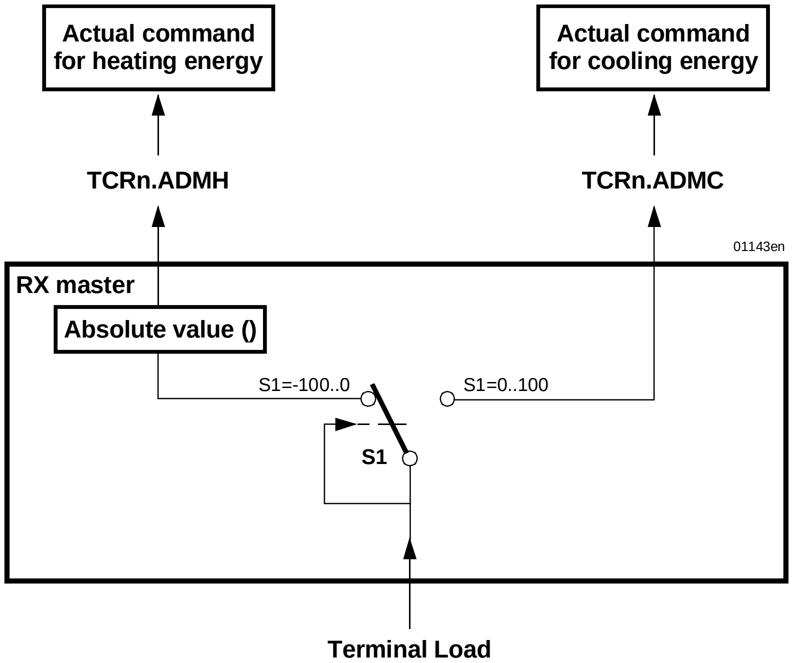
VISONIK BPS			NIDES.RX				RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*		*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.ASTH @TCRn.ASTC	Current sequence (see diagram)	←	nviUnitStatus . mode		1=HVAC_HEAT; 3=HVAC_COOL	←	**ne**	Unit Status	nvoUnitStatus	mode



3.2.20 Display current heating/cooling energy demand

All RXC applications

VISONIK BPS			NIDES.RX			RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.ADMC @TCRn.ADMH	See diagram	←	nviTerminalLoad	-100..0% = Heating 0..+100% = Cooling	←	**ne**	Terminal Load	nvoTerminalLoad



3.2.21 Room temperature averaging

All RXC applications

VISONIK BPS			NIDES.RX				RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.ARTP(x)	Current room temperature, controller x	←	nviTempSensor	°C	←	ne	Local Temp Sensor	nvoTempSensor nvoTempSensorPPS
		←	nviSpaceTemp	°C	←	1	Effect Space Temp	nvoSpaceTemp
@TCRn.RRTP	Room temperature average	→	nvoSpaceTemp	°C	→	ne	Space Temp	nviSpaceTemp
@TCRn.ROOMC=1	Room coordination active							

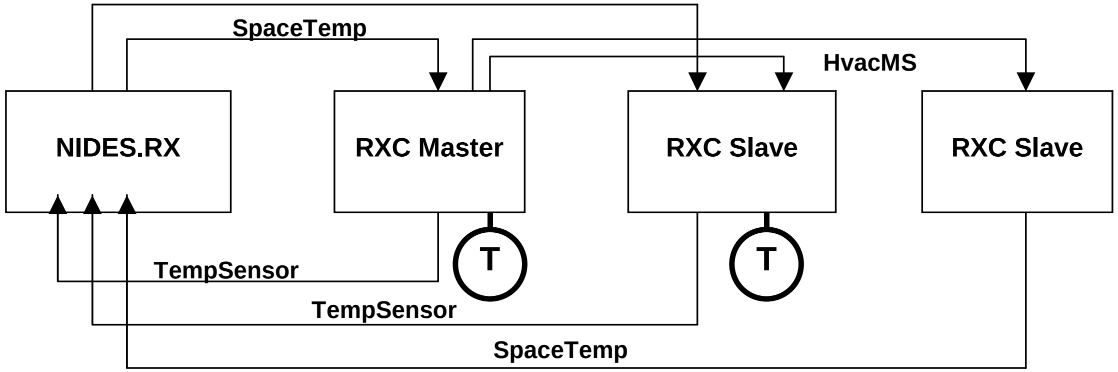
Use This function is used to specify the same room temperature in all controllers in the same room. It is used when a room point @TCR incorporates more than one controller, i.e. @TCR.TEC(m) > 0 for m = 1, 2, … The averaging function is a feature of the BPS TEC master.
It is only enabled if parameter is set to @TCR.ROOMC = 1.

ARTP selector logic The CFE software uses the following selector logic for the value of ARTP:
@TCR.ARTP(x) = nvoTempSensor (controller output), if nviSpaceTemp (controller input) is integrated
@TCR.ARTP(x) = nvoSpaceTemp (controller output), if nviSpaceTemp (controller input) is not integrated.

Bindings required In accordance with the selector logic, the following bindings are required:
For each controller with a local temperature sensor, bind NVs nvoTempSensor and nviSpaceTemp to the NIDES.RX

Sensor on room unit A distinction is made in Application Library V2 between a local room sensor on the controller (nvoTempSensor) and a room sensor on the operator unit (nvoTempSensorPPS). For integration into VISONIK, in the second case, a binding is required between the controller output nvo TempSensorPPS and the NIDES.RX input nviTempSensor.

Example The following diagram shows the bindings required for two controllers with and one controller without a local room sensor.



3.2.22 Definition of emergency mode (Emergency OFF / Smoke extraction)

All VAV applications, FPB05

VISONIK BPS			NIDES.RX				RXC controller	
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@AOx'42.EMV @TCRn.SOPST	See table of bindings	→	nvoEmergOverride	See table of bindings	→	ne	Emergency override	nviEmergOverride

Table of bindings

@AOx'42.EMV	@TCRn.SOPST		Emergeny Override	Description
x	5 = Smoke extraction	⇨	3 = EMERG_PURGE	All dampers fully open
0	6 = Emergency OFF	⇨	4 = EMERG_SHUTDOWN	All dampers closed
1	6 = Emergency OFF	⇨	1 = EMERG_PRESSURIZE	Supply air damper(s) fully open, extract air damper closed
2	6 = Emergency OFF	⇨	2 = EMERG_DEPRESSURIZE	Supply air damper(s) closed, extract air damper fully open

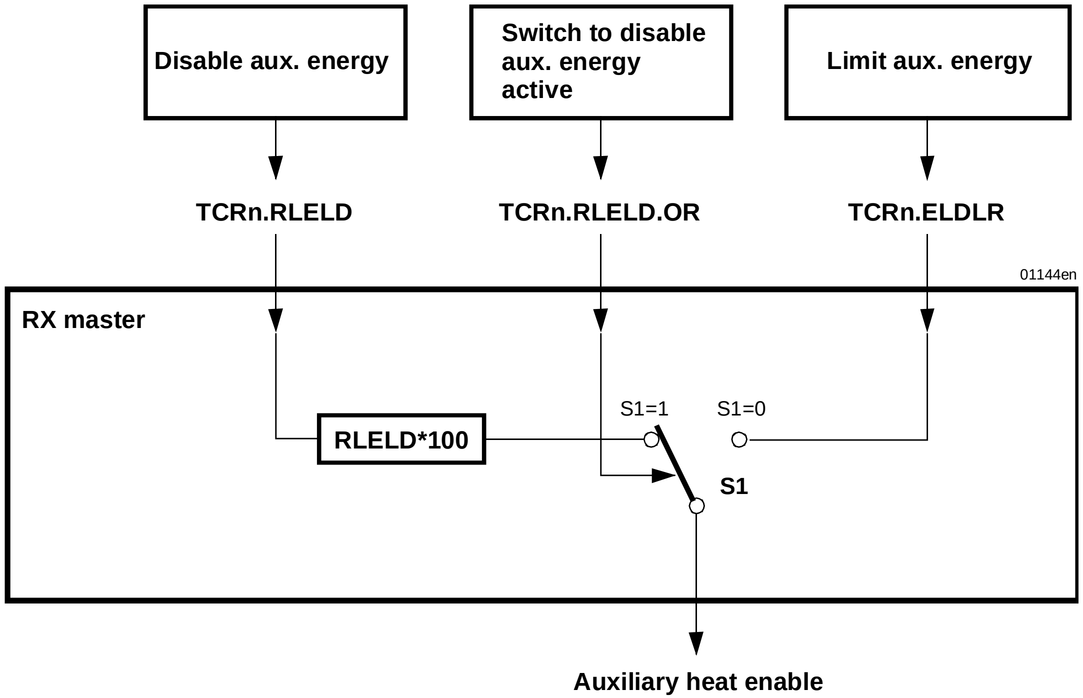
In operating state SOPST=6, Emergency Override is determined by link point @AOx'42. Single-parameter integration of iEmergOvr for the HVAC master in Room n with Controller number x). The value of must always be assigned before setting @TCRn.SOPST to 6.

Caution If the room concerned is subject to a time schedule(@DSTy), this must be deactivated during emergency operation (@DSTy.ACT=0), otherwise parameter @TCRn.SOPST will be overwritten again.

3.2.23 Disable or limit auxiliary energy supply

RAD03, FNC01, FNC03, VAV03, VAV06, FPB05

VISONIK BPS			NIDES.RX			RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@TCRn.RLELD @TCRn.RLELD.OR @TCRn.ELDLR	See diagram	→	nvoAuxHeatEnable . value	Enable auxiliary energy 0...100%	→	**ne**	Auxillary Heat Enable	nviAuxHeatEnable	Value



3.2.24 Intranet room operation

All RXC applications

VISONIK BPS			NIDES.RX				RXC controller	
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*
@TCRn.AOPST	See "Effective room status"	←	nviEffectOccup	See "Effective room status"	←	**1**	Effect Occup	nvoEffectOccup
@TCRy.ARTP	Current room temperature	←	nviSpaceTemp	Current room temperature	←	**1**	Effect Space Temp	nvoSpaceTemp
@AOx'24.EMV	Room status: 0 = Occupied 3 = Standby	→	nvoOccManCmd	Operating mode on room unit	→	*)	Occup Override	nviOccManCmd
@AOx'12.EMV	Sollwertkorrektur: -3 ... +3 °C	→	nvoSetptOffset	Setpoint adjustment from room unit	→	*)	Setpt Offset	nviSetptOffset
@AIx'80.EMV	Actual setpoint adjustment	←	nviSetptOffset	Actual setpoint adjustment	←	*)	Local Setpt Offset	nvoSetptOffset

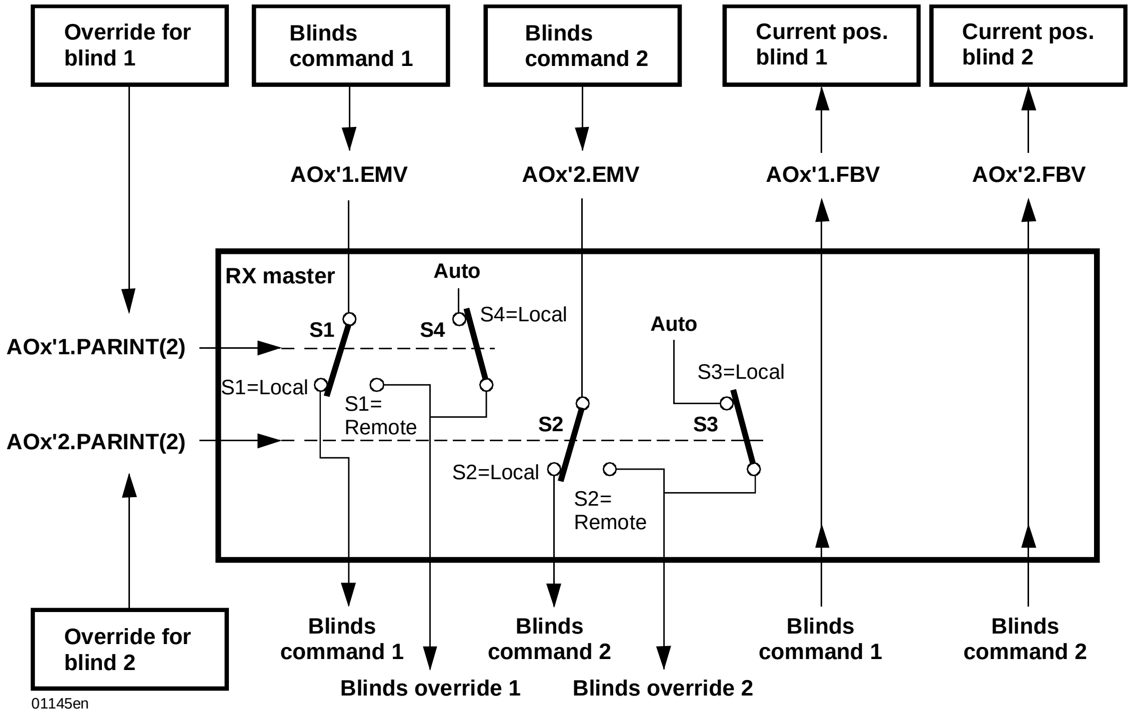
*) The Intranet room operation feature has its own standard binding template, which integrates these three additional points.

3.2.25 Control of blinds, 2 groups

INT01, INT02, INT03, INT04, INT06, INT07, INT15, INT17

VISONIK BPS			NIDES.RX			RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@AOx'1.EMV	Control of Blind 1	→	nvoSblndOverr_1	0=Up; 1=Down1; 2=Down2, 3=Auto	→	1	Blind Override #1	nviSblndOverr_1	
@AOx'1.PARINT(2)	(See diagram)	→	nvoSblndCmd_1	0 = Up; 1 = Down1; 2 = Down2	→	**ne**	Blind Command #1	nviSblndCmd_1	
@AOx'2.EMV	Control of Blind 2	→	nvoSblndOverr_2	0=Up; 1=Down1; 2=Down2, 3=Auto	→	1	Blind Override #2	nviSblndOverr_2	
@AOx'2.PARINT(2)	(See diagram)	→	nvoSblndCmd_2	0 = Up; 1 = Down1; 2 = Down2	→	**ne**	Blind Command #2	nviSblndCmd_2	
AOx'1.FBV	Current position of Blind 1	←	nviSblndCmd_1. setting	0 = Fully open... 100 = Fully closed	←	**1**	Blind Command #1	nvoSblndCmd_1	Setting
AOx'2.FBV	Current position of Blind 2	←	nviSblndCmd_2. setting	0 = Fully open... 100 = Fully closed	←	**1**	Blind Command #2	nvoSblndCmd_2	Setting

Ab1 = Jalousie geschlossen, Lamellen geöffnet
Ab2 = Jalousie geschlossen, Lamellen geschlossen



Note

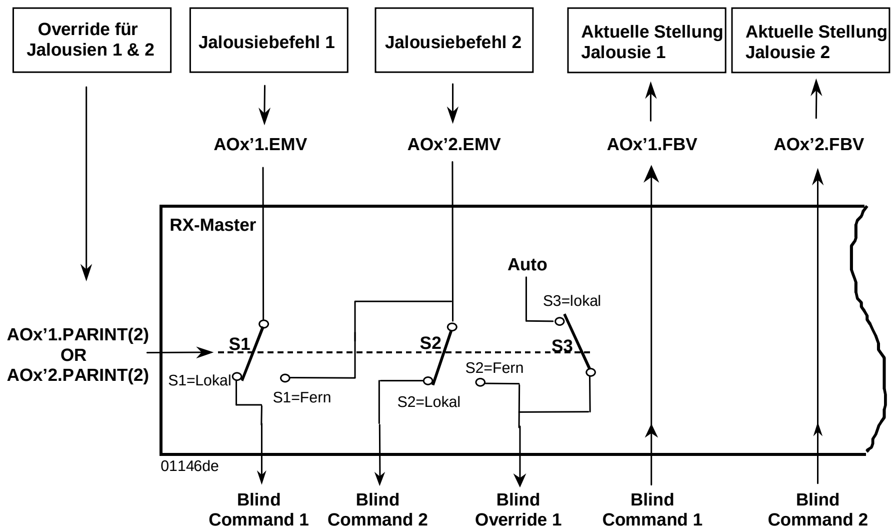
If the standard binding templates are used (nviSblndCmd_y) not integrated), the blinds are controlled in "Local" mode as follows:

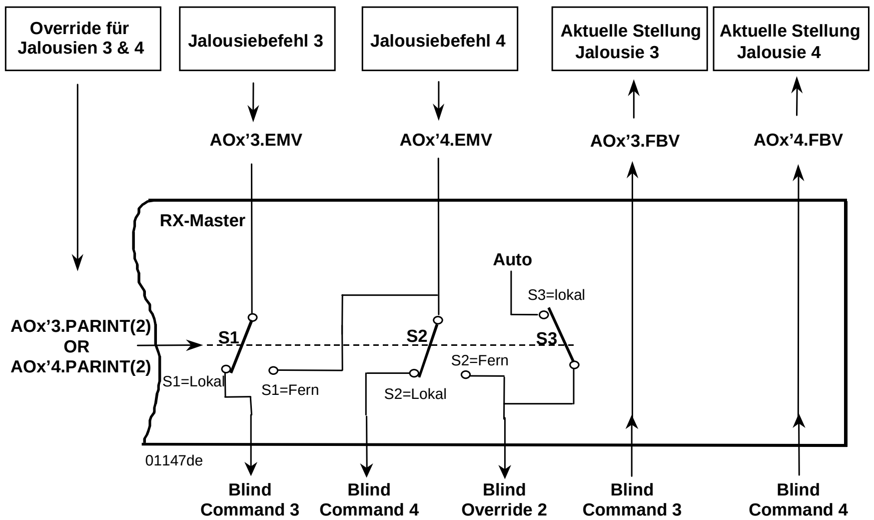
– If a new value is written to @Aox'y.EMV, the CFE software switches briefly to "Remote" and transmits the relevant command to the NV "Blind Override y".
– It then automatically reverts to "Local".

3.2.26 Control of blinds, 4 groups

INT05

VISONIK BPS			NIDES.RX				RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*		*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@AOx'1.EMV	Control of Blinds 1 and 2 (See diagram)	→	nvoSblndOverr_1		0=Up; 1=Down1; 2=Down2, 3=Auto	→	1	Blind Override #1	nviSblndOverr_1	
@AOx'1.PARINT(2)		→	nvoSblndCmd_1		0 = Up; 1 = Down1; 2 = Down2	→	**ne**	Blind Command #1	nviSblndCmd_1	
@AOx'2.EMV		→	nvoSblndCmd_2		0 = Up; 1 = Down1; 2 = Down2	→	**ne**	Blind Command #2	nviSblndCmd_2	
@AOx'2.PARINT(2)										
@AOx'3.EMV	Control of Blinds 3 and 4 (See diagram)	→	nvoSblndOverr_2		0=Up; 1=Down1; 2=Down2, 3=Auto	→	1	Blind Override #2	nviSblndOverr_2	
@AOx'3.PARINT(2)		→	nvoSblndCmd_3		0 = Up; 1 = Down1; 2 = Down2	→	**ne**	Blind Command #3	nviSblndCmd_3	
@AOx'4.EMV		→	nvoSblndCmd_4		0 = Up; 1 = Down1; 2 = Down2	→	**ne**	Blind Command #4	nviSblndCmd_4	
@AOx'4.PARINT(2)										
AOx'1.FBV	Current position of Blind 1	←	nviSblndCmd_1 . setting		0 = Fully open... 100 = Fully closed	←	**1**	Blind Command #1	nvoSblndCmd_1	Setting
AOx'2.FBV	Current position of Blind 2	←	nviSblndCmd_2 . setting		0 = Fully open... 100 = Fully closed	←	**1**	Blind Command #2	nvoSblndCmd_2	Setting
AOx'3.FBV	Current position of Blind 3	←	nviSblndCmd_3 . setting		0 = Fully open... 100 = Fully closed	←	**1**	Blind Command #3	nvoSblndCmd_3	Setting
AOx'4.FBV	Current position of Blind 4	←	nviSblndCmd_4 . setting		0 = Fully open... 100 = Fully closed	←	**1**	Blind Command #4	nvoSblndCmd_4	Setting

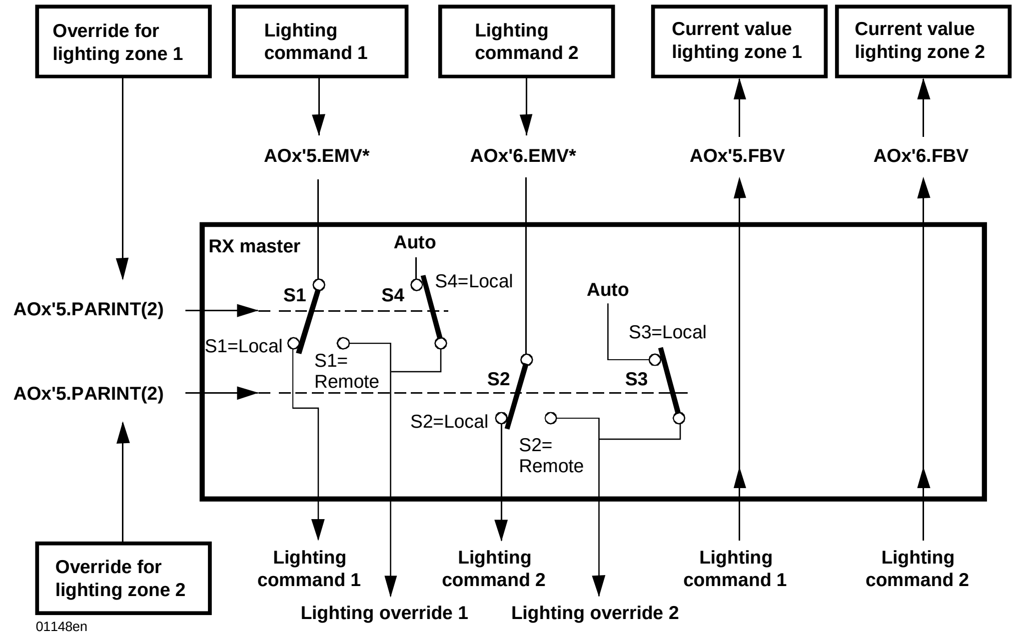




3.2.27 Lighting control, 2 groups

INT01, INT04, INT05, INT10, INT12, INT15, INT17

VISONIK BPS			NIDES.RX				RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@AOx'5.EMV	Control of lighting group 1 (see diagram)	→	nvoLightOverr_1	0 = OFF; 1 = ON, 255 = Auto	→	1	Light Override #1	nviLightOverr_1	
@AOx'5.PARINT(2)		→	nvoLightCmd_1	0 = Off, 0.5...100 = On, n%	→	**ne**	Light Command #1	nviLightCmd_1	
@AOx'6.EMV	Control of lighting group 2 (see diagram)	→	nvoLightOverr_2	0 = OFF; 1 = ON, 255 = Auto	→	1	Light Override #2	nviLightOverr_2	
@AOx'6.PARINT(2)		→	nvoLightCmd_2	0 = Off, 0.5...100 = On, n%	→	**ne**	Light Command #2	nviLightCmd_2	
AOx'5.FBV	Current lighting value, Gp 1	←	nviLightCmd_1. value	0 = Off, 0.5...100 = On, n%	←	**1**	Light Command #1	nvoLightCmd_1	Value
AOx'6.FBV	Current lighting value, Gp 2	←	nviLightCmd_2. value	0 = Off, 0.5...100 = On, n%	←	**1**	Light Command #2	nvoLightCmd_2	Value



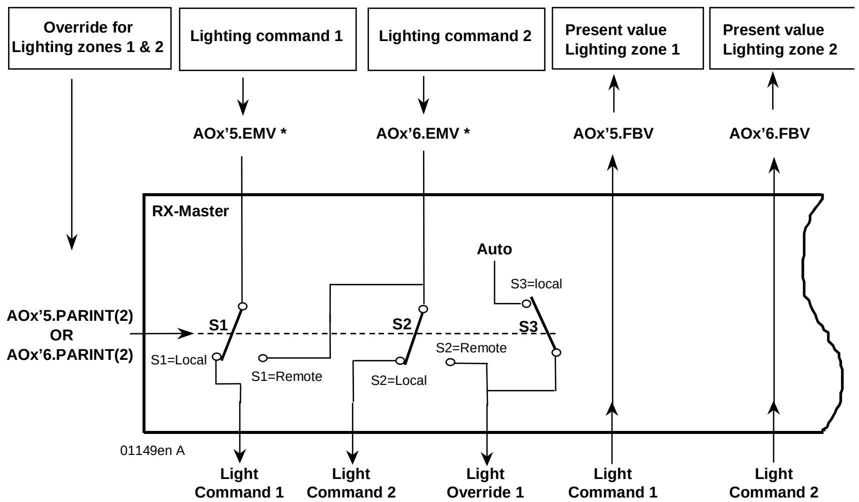
Note
If the standard binding templates are used (nviLightCmd_y) not integrated), the lighting is controlled in "Local" mode as follows:
f a new value is written to @AOx'y.EMV, the CFE software switches briefly to "Remote" and transmits the relevant command to the NV "Light Override y". It then reverts automatically to "Local".

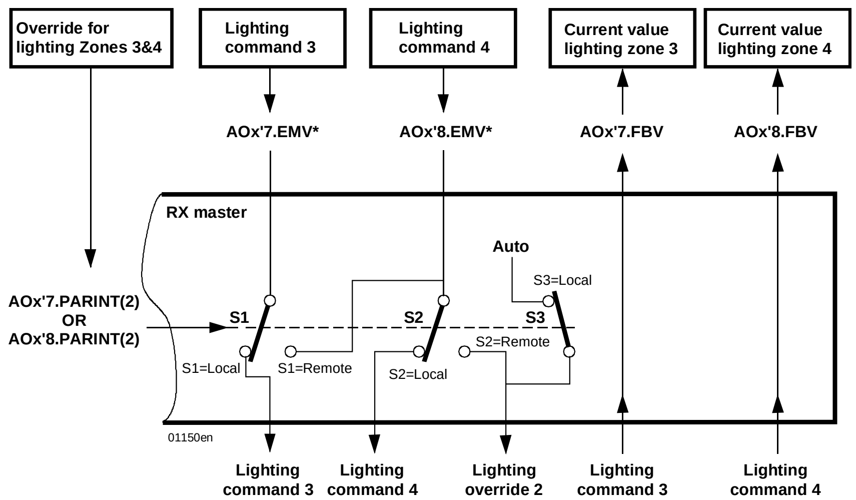
.

3.2.28 Lighting control, 4 groups

INT02, INT03, INT06, INT07, INT11

VISONIK BPS			NIDES.RX			RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@AOx'5.EMV	Control of lighting groups 1 and 2 (see diagram)	→	nvoLightOverr_1	0 = OFF; 1 = ON, 255 = Auto	→	1	Light Override #1	nviLightOverr_1	
@AOx'5.PARINT(2)		→	nvoLightCmd_1	0 = Off, 0.5...100 = On, n%	→	**ne**	Light Command #1	nviLightCmd_1	
@AOx'6.EMV		→	nvoLightCmd_2	0 = Off, 0.5...100 = On, n%	→	**ne**	Light Command #2	nviLightCmd_2	
@AOx'6.PARINT(2)									
@AOx'7.EMV	Control of lighting groups 3 and 4 (See diagram)	→	nvoLightOverr_2	0 = OFF; 1 = ON, 255 = Auto	→	1	Light Override #2	nviLightOverr_2	
@AOx'7.PARINT(2)		→	nvoLightCmd_3	0 = Off, 0.5...100 = On, n%	→	**ne**	Light Command #3	nviLightCmd_3	
@AOx'8.EMV		→	nvoLightCmd_4	0 = Off, 0.5...100 = On, n%	→	**ne**	Light Command #4	nviLightCmd_4	
@AOx'8.PARINT(2)									
AOx'5.FBV	Current lighting value Gp1	←	nviLightCmd_1. value	0 = Off, 0.5...100 = On, n%	←	**1**	Light Command #1	nvoLightCmd_1	Value
AOx'6.FBV	Current lighting value Gp2	←	nviLightCmd_2 . value	0 = Off, 0.5...100 = On, n%	←	**1**	Light Command #2	nvoLightCmd_2	Value
AOx'7.FBV	Current lighting value Gp3	←	nviLightCmd_3 . value	0 = Off, 0.5...100 = On, n%	←	**1**	Light Command #3	nvoLightCmd_3	Value
AOx'8.FBV	Current lighting value Gp4	←	nviLightCmd_4 . value	0 = Off, 0.5...100 = On, n%	←	**1**	Light Command #4	nvoLightCmd_4	Value





3.2.29 Additional single parameters for HVAC control

Depends on application

VISONIK BPS			NIDES.RX				RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Function/Range*		BT	*User designation*	*NV name*	*NV element*
@AOx'19.EMV	Volume flow setpoint	→	nvoAirflowSetpt	l/s	→	ne	Aiflow Setpoint	nviAirflowSetpt	
@AOx'25.EMV	Occupancy sensor	→	nvoOccSensor	0 = Occupied, 1 = Unocc.	→	ne	Occup Sensor	nviOccSensor	
@AOx'27.EMV	Definition of sequence: heating/cooling	→	nvoHeatCool	0 = Auto, 1 = Heat, 3 = Cool	→	ne	Heat Cool Mode	nviHeatCool	
@AOx'29.EMV	Definition of fan speed	→	nvoFanSpeedCmd	0 = Off, 1 = St.1, 2 = St. 2, 3 = St.3, 4 = Auto	→	ne	Fan Speed Command	nviFanSpeedCmd	
@AOx'33.EMV	Economizer enable (fan coil)	→	nvoEconEnable . state	0 = OFF, 1 = ON	→	ne	Economiser Enable	nviEconEnable	State
@AOx'35.EMV	Energy hold-off	→	nvoEnergyHoldOff . state	0 = OFF, 1 = ON	→	ne	Energy Hold-off	nviEnergyHoldOff	State
@AOx'43.EMV	Temperature of primary medium	→	nvoSourceTemp	°C	→	ne	Source Temp	nviSourceTemp	
@AOx'44.EMV	Outdoor temperature	→	nvoOutdoorTemp	°C	→	ne	Outdoor Temp	nviOutdoorTemp	
@AOx'45.EMV	Indoor air humidity	→	nvoSpaceRH	% rh	→	ne	Space Humidity	nviSpaceRH	
@AOx'46.EMV	CO_2 concentration in room	→	nvoSpaceCO2	ppm	→	ne	Space CO2	nviSpaceCO2	
@AOx'47.EMV	Dewpoint temperature of indoor air	→	nvoSpaceDewPt	°C	→	ne	Space Dewpoint	nviSpaceDewPt	
@AOx'48.EMV	Dewpoint temp. of outdoor air	→	nvoOutdoorDewPt	°C	→	ne	Outdoor Dewpoint	nviOutdoorDewPt	
@AOx'53.EMV	Temp. of primary heating medium	→	nvoSourceTempH	°C	→	ne	Source Temp Heating	nviSourceTempH	
@AOx'68.EMV	Min. outside air damper position	↔	ncoOAMinPos	%	↔	ne	Outs. Air Dmp Min Pos	nciOAMinPos	
@AIx'13.EMV	CO_2 concentration in room	←	nviSpaceCO2	ppm	←	ne	Space CO2	nvoSpaceCO2	
@AIx'42.EMV	Current setpoint	←	nviEffectSetpt	°C	←	ne	Effect Setpt	nvoEffectSetpt	
@AIx60'.EMV	Supply air temperature	←	nviDischAirTemp	°C	←	ne	Discharge Air Temp	nvoDischAirTemp	
@AIx'66.EMV	Position of outside air damper	←	nviOADamper	0 ... 100 %	←	ne	Outdoor Air Damper	nvoOADamper	
@AIx'67.EMV	Indoor air humidity	←	nviSpaceRH	% rh	←	ne	Space Humidity	nvoSpaceRH	
@AIx'68.EMV	Dew point monitoring	←	nviSpaceDewPt	50°C = Normal, -10°C = Condensation	←	ne	Space Dewpoint	nvoSpaceDewPt	
@AIx'81.EMV	Local occupancy sensor	←	nviOccSensor	0 = Unoccupied 1 = Occupied	←	ne	Local Occup Sensor	nvoOccSensor	
@AIx'82.EMV	Volume flow temperature	←	nviFlowTemp	°C	←	ne	Flow Temp	nvoFlowTemp	
@AIx'83.EMV	Pump status	←	nviPumpStatus. value	0 = Off, 0.5...100% = On	←	ne	Pump Status	nvoPumpStatus	Value

3.2.30 Other single parameters for lighting control

All INT applications

VISONIK BPS			NIDES.RX			RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*	*Value range*		BT	*User designation*	*NV name*
@AOx'111.EMV	Measured daylight intensity	→	nvoLightLuxLevel	Lux	→	ne	Daylight Lux Level	nviLightLuxLevel
@AOx112'.EMV	Daylight switch-off threshold 1	↔	ncoLuxThrsldOff_1	Lux	↔	ne	Daylight Threshold #1	nciLuxThrsldOff_1
@AOx113'.EMV	Daylight switch-off threshold 2	↔	ncoLuxThrsldOff_2	Lux	↔	ne	Daylight Threshold #1	nciLuxThrsldOff_2

3.2.31 Single parameters for I/O functions

Appilcations 000xx

VISONIK BPS			NIDES.RX			RXC controller			
DP.Parameter	*Function/Range*		*NV name / element*	*Value range*		BT	*User designation*	*NV name*	*NV element*
@AOx'133.EMV	Controller output Y1	→	nvoCtrl_Y1 . state	0 = Off, 1 = On	→	ne	Controller Y1	nviCtrl_Y1	State
@AOx'135.EMV	Controller output Y2	→	nvoCtrl_Y2 . state	0 = Off, 1 = On	→	ne	Controller Y2	nviCtrl_Y2	State
@AOx'137.EMV	Controller output Y3	→	nvoCtrl_Y3 . state	0 = Off, 1 = On	→	ne	Controller Y3	nviCtrl_Y3	State
@AOx'139.EMV	Controller output Y4	→	nvoCtrl_Y4 . state	0 = Off, 1 = On	→	ne	Controller Y4	nviCtrl_Y4	State
@AOx'20.EMV	Controller output Y5	→	nvoCtrl_Y5 . state	0 = Off, 1 = On	→	ne	Controller Y5	nviCtrl_Y5	State
@AOx'36.EMV	Controller output Y6	→	nvoCtrl_Y6 . state	0 = Off, 1 = On	→	ne	Controller Y6	nviCtrl_Y6	State
@AOx'141.EMV	Controller output Q14	→	nvoCtrl_Q14 . state	0 = Off, 1 = On	→	ne	Controller Q14	nviCtrl_Q14	State
@AOx'143.EMV	Controller output Q24	→	nvoCtrl_Q24 . state	0 = Off, 1 = On	→	ne	Controller Q24	nviCtrl_Q24	State
@AOx'145.EMV	Controller output Q34	→	nvoCtrl_Q34 . state	0 = Off, 1 = On	→	ne	Controller Q34	nviCtrl_Q34	State
@AOx'147.EMV	Extension module 1, output Q14	→	nvoExMod_1_Q14 . state	0 = Off, 1 = On	→	ne	Ext Mod #1 Q14	nviExMod_1_Q14	State
@AOx'149.EMV	Extension module 1, output Q24	→	nvoExMod_1_Q24 . state	0 = Off, 1 = On	→	ne	Ext Mod #1 Q24	nviExMod_1_Q24	State
@AOx'151.EMV	Extension module 1, output Q34	→	nvoExMod_1_Q34 . state	0 = Off, 1 = On	→	ne	Ext Mod #1 Q34	nviExMod_1_Q34	State
@AOx'153.EMV	Extension module 1, output Q44	→	nvoExMod_1_Q44 . state	0 = Off, 1 = On	→	ne	Ext Mod #1 Q44	nviExMod_1_Q44	State
@AOx'155.EMV	Extension module 2, output Q14	→	nvoExMod_2_Q14 . state	0 = Off, 1 = On	→	ne	Ext Mod #2 Q14	nviExMod_2_Q14	State
@AOx'157.EMV	Extension module 2, output Q24	→	nvoExMod_2_Q24 . state	0 = Off, 1 = On	→	ne	Ext Mod #2 Q24	nviExMod_2_Q24	State
@AOx'159.EMV	Extension module 2, output Q34	→	nvoExMod_2_Q34 . state	0 = Off, 1 = On	→	ne	Ext Mod #2 Q34	nviExMod_2_Q34	State
@AOx'161.EMV	Extension module 2, output Q44	→	nvoExMod_2_Q44 . state	0 = Off, 1 = On	→	ne	Ext Mod #2 Q44	nviExMod_2_Q44	State
@AIx'153.EMV	Controller analog input B1	←	nviCtrl_B1	°C	←	ne	Controller B1	nvoCtrl_B1	

VISONIK BPS			NIDES.RX					RXC controller		
DP.Parameter	*Function/Range*		*NV name / element*		*Value range*		BT	*User designation*	*NV name*	*NV element*
@AIx'155.EMV	Controller input D1	←	nviCtrl_D1 . state		0 = OFF; 1 = ON	←	ne	Controller D1	nvoCtrl_D1	State
@AIx'157.EMV	Controller input D2	←	nviCtrl_D2 . state		0 = OFF; 1 = ON	←	ne	Controller D2	nvoCtrl_D2	State
@AIx'159.EMV	Controller input D3	←	nviCtrl_D3 . state		0 = OFF; 1 = ON	←	ne	Controller D3	nvoCtrl_D3	State
@AIx'161.EMV	Controller input D4	←	nviCtrl_D4 . state		0 = OFF; 1 = ON	←	ne	Controller D4	nvoCtrl_D4	State
@AIx'163.EMV	Extension module 1, input D1	←	nviExMod_1_D1 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #1 D1	nvoExMod_1_D1	State
@AIx'165.EMV	Extension module 1, input D2	←	nviExMod_1_D2 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #1 D2	nvoExMod_1_D2	State
@AIx'167.EMV	Extension module 1, input D3	←	nviExMod_1_D3 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #1 D3	nvoExMod_1_D3	State
@AIx'169.EMV	Extension module 1, input D4	←	nviExMod_1_D4 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #1 D4	nvoExMod_1_D4	State
@AIx'171.EMV	Extension module 2, input D1	←	nviExMod_2_D1 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #2 D1	nvoExMod_2_D1	State
@AIx'173.EMV	Extension module 2, input D2	←	nviExMod_2_D2 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #2 D2	nvoExMod_2_D2	State
@AIx'175.EMV	Extension module 2, input D3	←	nviExMod_2_D3 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #2 D3	nvoExMod_2_D3	State
@AIx'177.EMV	Extension module 2, input D4	←	nviExMod_2_D4 . state		0 = OFF; 1 = ON	←	ne	Ext Mod #2 D4	nvoExMod_2_D4	State

3.3 Definition of HVAC groups

Introduction

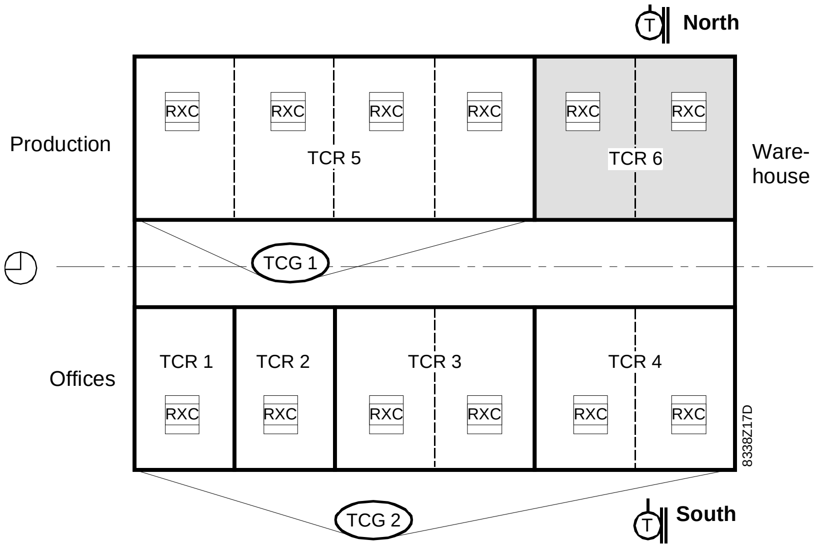
HVAC groups are defined to take account of various HVAC and organizational aspects affecting the building, such as:

– The direction in which the building faces (HVAC zones / groups based on HVAC requirements)
– The use of a number of rooms at given times (groups based on occupancy pattern)

The room as the basic element for RXC integration

In the context of the integration of RXC into VISONIK, the basic element for the definition of HVAC groups is the room. The diagram below is an example of a floor plan to which the following conditions apply:

– The north and south facing sides of the building each consist of 6 modules, each comprising an RXC controller and the associated field devices. The corresponding TEC rooms (master/slave relationships) are defined on the basis of these modules (master/slave relationships).
– The offices and the production area are in different HVAC zones, but share the same occupancy times (07:30 to 18:00).
– The warehouse is maintained at a constant temperature round the clock.



Notes on the diagram

For the example above, “TEC rooms” and HVAC groups are defined as follows:

Name	Element	Details
TCR 1 .. 4	TEC Rooms 1 … 4	Offices: Comfort setpoints, heating/cooling 18 °C / 24 °C
TCR 5	TEC Room 5	Production area: Comfort setpoints, heating/cooling 21 °C / 24 °C
TCR 6	TEC Room 6	Warehouse: Constant 20 °C, no time program
TCG1	HVAC Group 1	North-facing zone: Summer compensation
TCG 2	HVAC Group 2	South-facing zone: Summer compensation

Notes on defining groups

Note the following when defining HVAC groups:

– Only rooms can be added to HVAC groups.
– Max. 10 RXC controllers per TEC room, and max. 20 TEC rooms per HVAC group.
– A room may not be assigned to more than one HVAC group.

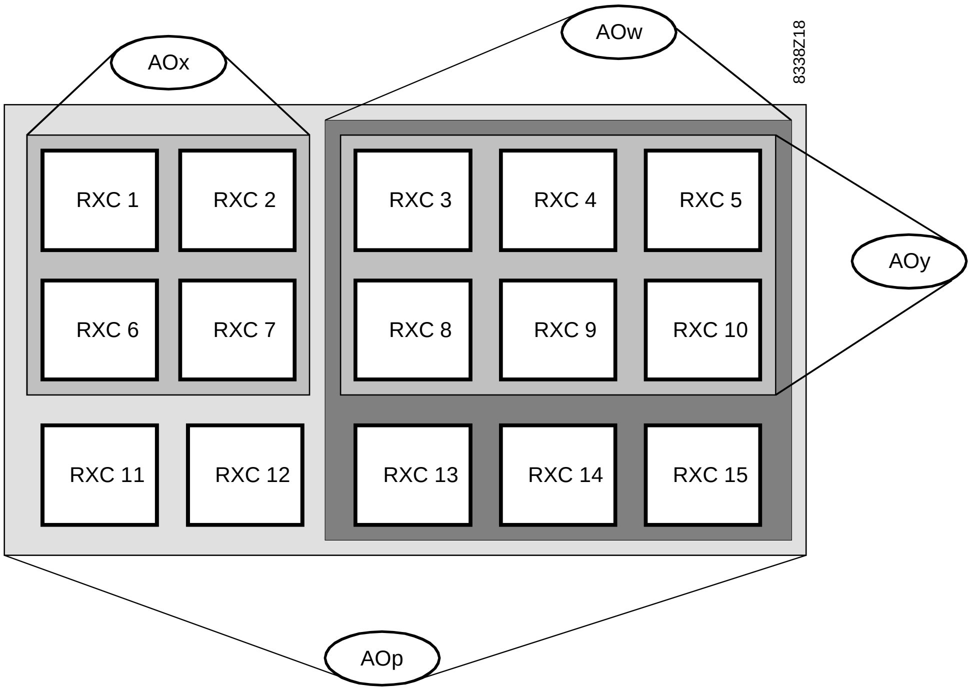
3.4 Defining lighting and blind groups

Introduction

In addition to HVAC functions, the integrated applications of DESIGO RXC include the control of lighting and blinds. This makes it necessary to define groups for these purposes too, whether in response to HVAC conditions or differing patterns of room occupancy.

The RXC controller as the basic element

Whereas the room is the basic element for defining HVAC groups, in the case of blinds and lighting, the definition of groups is based on the RXC controller. The diagram below shows the principles which apply to the definition of blind or lighting groups.



Notes on the diagram

Four groups have been defined, covering the complete area:

Group	Covering…
AOp	All rooms
AOw	A subset of AOp
AOx	A subset of AOp
AOy	A subset of AOw

Note: p .. y can be numbers in the range 1 .. 200.

Rules for the definition of groups

In summary, the following rules apply to the definition of lighting and blind groups:

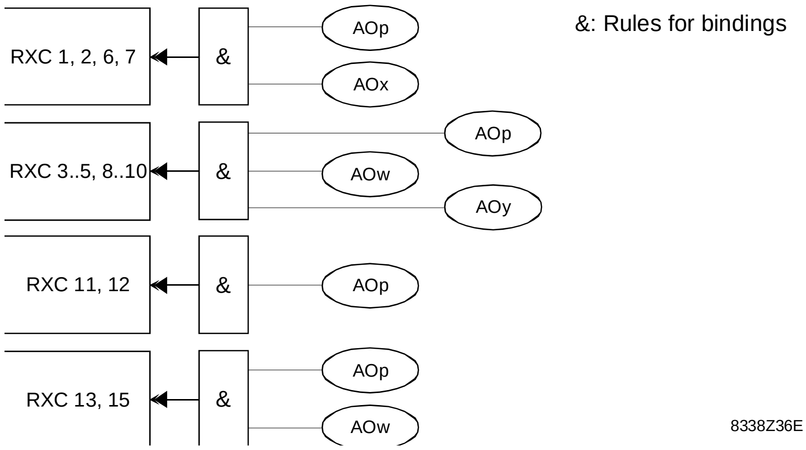
- The RXC controller is the basic element for the definition of groups.
- The window-blind control functions of the RXC controller can be assigned to several blind groups simultaneously.
- The lighting control functions of the RXC controller can be assigned to several lighting groups simultaneously.
- The smallest unit is the controller, i.e. the lighting zones or blinds connected to a controller always belong to the same group(s).

3.5 Control priority functions for lighting and blind groups

Problem

Since one RXC controller can be assigned to a number of lighting or blind groups, each with different group values, a method must be found to determine which group is to act on the RXC controller. The diagram below highlights the problem on the basis of the groupings illustrated earlier.

If, for example, a command applies to group Aop, this will transmit the command to the associated RXC controllers, including RXC controllers 1, 2, 6 and 7. At the same time, however, group AOx also acts on these controllers. How is this resolved?



Solution

The problem described above is solved as follows:

- Assign priorities to the blind and lighting groups.
- When a controller belongs to several groups of the same priority, the last command always takes precedence.

3.6 Examples for lighting and blind groups

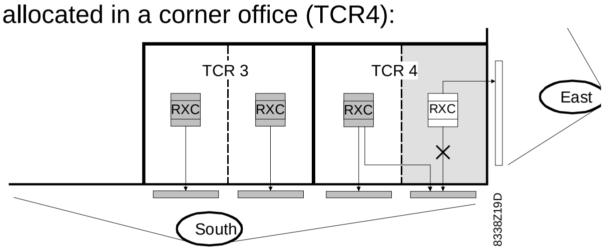
Introduction

This page shows two typical examples of the application of lighting and blind groups. These examples illustrate:

– Why the RXC controller has to be treated as the basic unit for lighting and blind groups
– How overlapping groups occur and how these are handled

Blinds in corner offices

The diagram below shows how the blind outputs are connected, and how groups are allocated in a corner office (TCR4):



Notes on the diagram

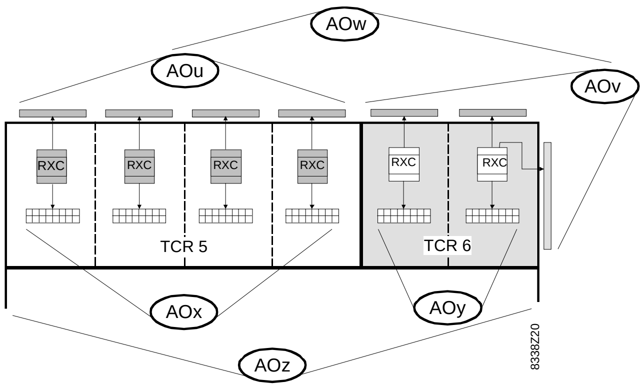
The groups defined in the diagram above have the following characteristics:

Group	Description
South	All blinds on the south-facing façade form a group which is operated remotely via the system (e.g. “Down” at night, “Up” in the event of a storm).
East	In the corner office, the blind on the east façade is disconnected from the “South” group and assigned to the “East” group, so that it provides shade only when required.

Overlapping groups

The diagram below shows two rooms with blinds and lighting zones. Assumptions:

– TCR5, Production area: blinds and lighting for the entire production area, operated by manual switches
– TCR 6, Warehouse: blinds normally down, lighting only on when required (occupancy sensor)



Notes on the diagram

Further, the following groups are defined for operation via the system:

Group	Description
AOu	Blind group for Production area
AOv	Blind group for Warehouse
AOw	Blind group for whole building façade
AOx	Lighting group for Production area
AOy	Lighting group for Warehouse
AOz	Lighting group for entire floor: “Off” at night, via command from system

4 Engineering
4.1 Overview

Introduction

This section concerns the engineering required to integrate the DESIGO RXC system into VISONIK BPS. The key areas covered are:

– Data flow from the plant to the VISONIK data points
– Engineering procedure
– The configuration of groups for HVAC, blinds and lighting
– Single-parameter integration

Contents of this section

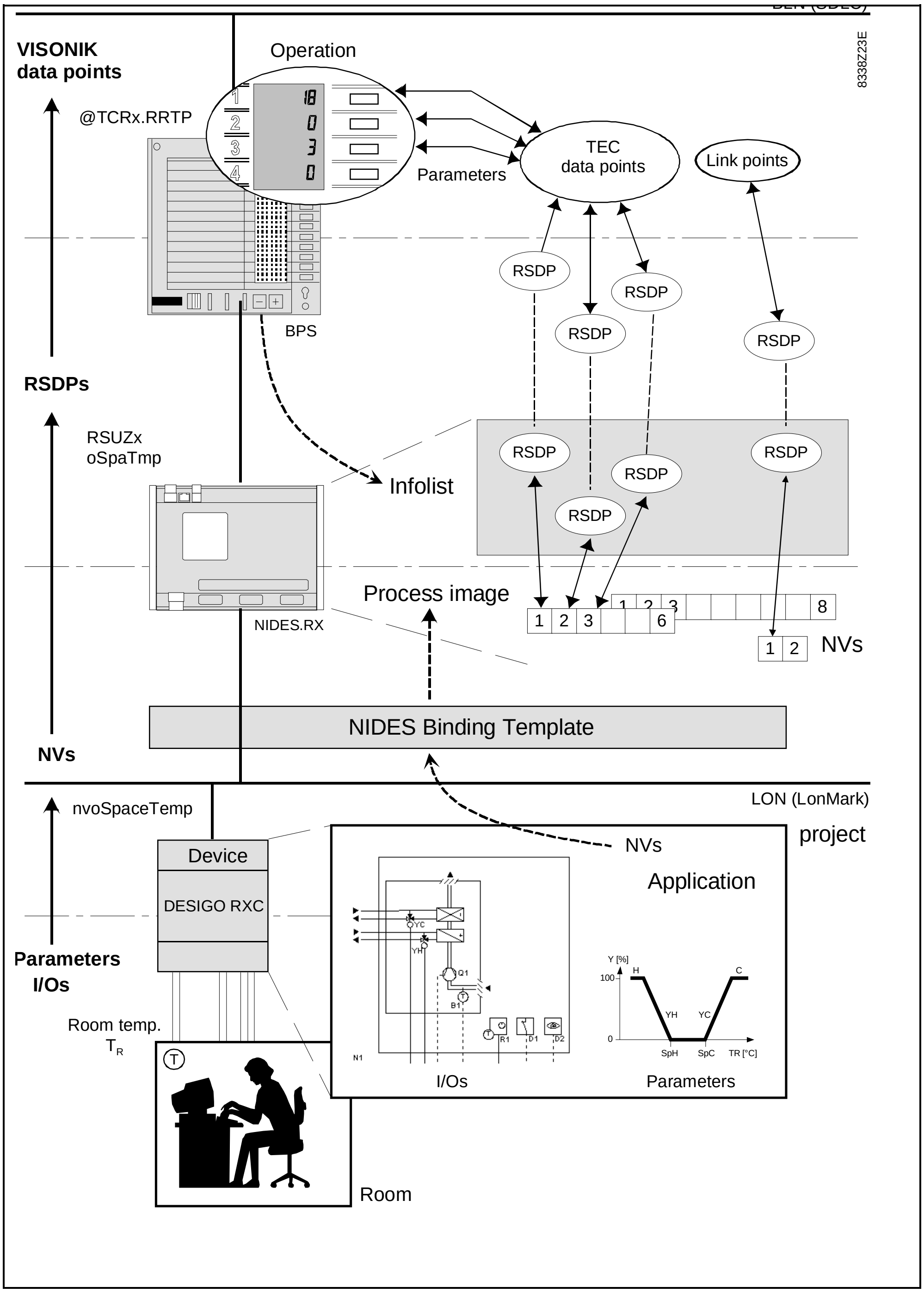
4.2 From the application to the TEC data points

Purpose of the overview

The diagram below is a simplified illustration of the route taken by data from the I/Os and parameters of the application through to the VISONIK data points in the BPS. Its purpose is

– To provide an overall picture of the data flow, from the perspective of the application
– To illustrate the interaction between the various elements.

This will help clarify the engineering steps described later.



Note

The data flow is indicated in the left of the diagram from the bottom to the top, using a room temperature T_R as an example. Data is, of course, also transmitted from the top down (e.g. setpoints).

Description of individual elements

The individual elements in the diagram on the previous page are as follows (starting at the bottom):

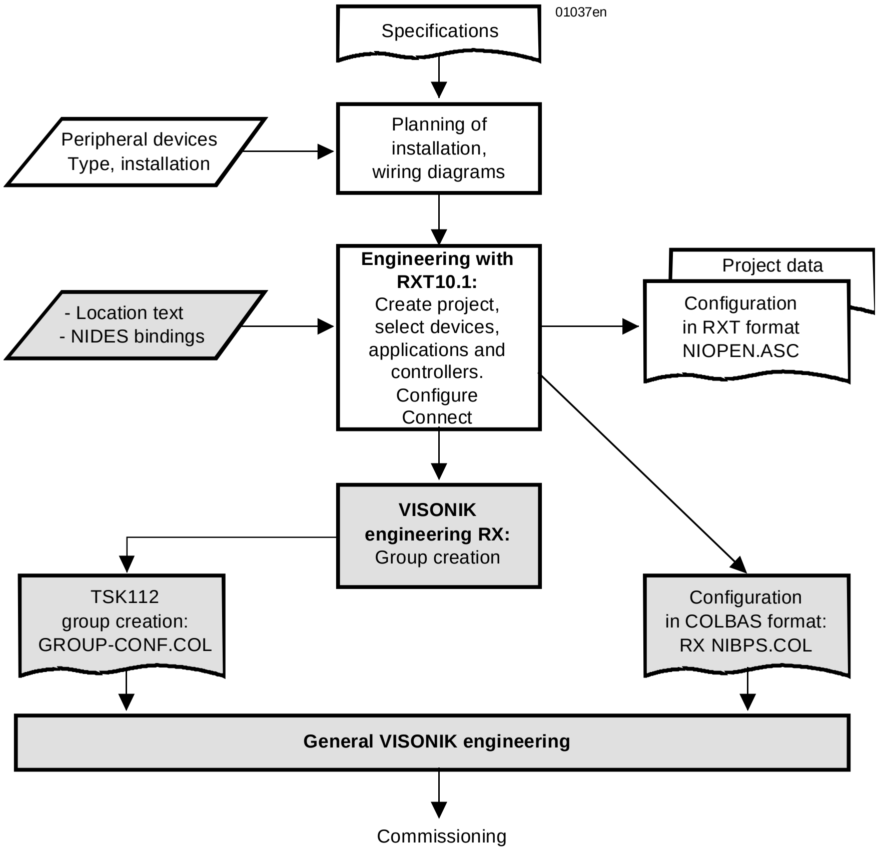
Element	Description
Application	The term “application” refers to the software solution for a given HVAC application, e.g. a room fan-coil system. The application maps all inputs, outputs and parameters in the “device”.
Device	The “device” refers to the type of DESIGO RXC device in the database of the RXT10.1 tool, and is a component of a project.
Project	The “project” reflects the structure of a LON network. It contains all the information about the existing devices and their bindings.
NVs	DESIGO RX converts the parameter and I/O information into network variables – the language used by the LON. Each NV has a name, and comprises one or more elements. Examples: *Name* – *Description* – *Elements* nvoSpaceTemp – Actual room temperature – 1 nciSetpoints – Definition of setpoints for heating/cooling – 6 For nciSetpoints, for example, the six elements represent the setpoints for heating/cooling in the various operating modes.
NIDES binding template	The NIDES binding templates determine which network variables are adopted by the NIDES.RX. For this purpose, the RXT10 commissioning and service tool includes two standard templates for each application. *Note:* The NIDES binding templates should not be confused with the templates used for bindings between the RXC controllers.
Process image	The process image in the NIDES.RX contains all the information obtained from the “project” on the basis of all NIDES binding templates.
RSDP	Abbreviation for “RS data point”. Unit of information in the AS1000 and MS2000 systems. The NV elements are exchanged between the BPS and NIDES.RX in the form of RSDPs of a particular type. The RSDP for a given NV is identified by the “SAPIM” text labels. This mapping information is stored in the NIOPEN.ASC file or in the RxNiBps.col configuration file for the BPS. *Example:* The NV nvoSpaceTemp is transferred as an RSDP of the type RSUZ. The associated SAPIM text label is oSpaTmp.
Infolist	The Infolist contains the RSDPs for all network variables intended for integration into the system. It is downloaded from the BPS into the NIDES.RX at the commissioning stage.
VISONIK data points	The network variables associated with the RSDPs are mapped to VISONIK parameters (TEC points or link points), cf. Section 3.2. *Example:* The value of nvoSpaceTemp goes to parameter @TCR.RRTP.
Operation	The VISONIK parameters can be operated on the BPS via: - VISOTOOL Editor / COLBAS - the POP cards (not available for link point parameters)

4.3

4.4 Overall engineering procedure

Flow chart

The flow chart below is a simplified illustration of the complete engineering process for the integration of the DESIGO RXC controllers. Activities and elements related specifically to VISONIK are shown on a shaded background.



VISONIK-specific activities

The table below contains brief details of the activities specifically related to VISONIK:

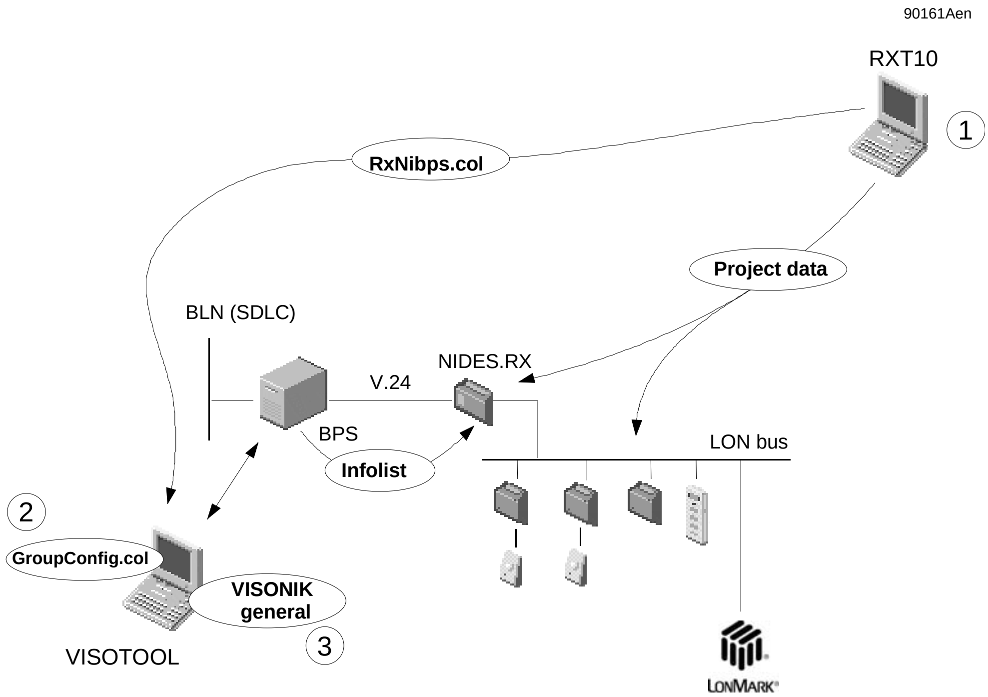
Activity	Description
Engineering with the RXT10	The procedure for engineering with the RXT10 commissioning and service tool is as described in manual CA2B3808E. It is important to note the following, however: – the VISONIK-specific allocation of the location text – the use of NIDES binding templates VISONIK 1 / 2
VISONIK RXC engineering	The VISONIK RXC engineering comprises: – *Definition of groups:* Definition of HVAC, lighting and blind groups by editing TSK 122 (if available; otherwise it must be created)
General VISONIK engineering	The general VISONIK engineering consists of: – Downloading the RXC coupling software into the BPS – Configuration of group and room functions in the TEC data points, e.g.: Summer/winter compensation, room temperature monitoring, optimum start program, text commentary, etc.

The VISONIK-specific activities associated with RXC integration are described in more detail in the workflow section below and under the subsequent sub-headings.

4.5 Workflow for VISONIK RXC integration

Overview

The diagram below shows the workflow for the integration of DESIGO RXC into VISONIK.



Stage-by-stage description

The various stages in the workflow diagram involve the following activities and results:

Stage	Activities/Outcome
1	Creation of the project (room level) with the RXT10, including: Here: – Entry of location text for the devices in compliance with VISONIK addressing. – Use of NIDES binding templates VISONIK 1 and 2. *Outcome:* When saving the projects, the RXT10 creates the following files: – "Project data": Image of the relevant LON network with all devices, applications, parameters and connections. – "RxNibps.col." export file for the BPS, comprising: – TSK111 : Project image for BPS – TSK110 : Infolist for NIDES.RX
2	Definition of groups: "GroupConfig.col" consisting of: TSK 112 data for HVAC, blind and lighting groups. This data should be adapted specifically to each project with the VISOTOOL Editor.
3	VISONIK engineering of the BPS: – Loading of CFE software CFE_V2_x_y.col – Loading of project data RxNiBps.col and GroupConfig.col – Other VISONIK engineering (TEC master functions, time program, text commentary and additional tasks and points.

At the commissioning stage, project data is loaded onto the LON network with the RXT10 tool, and VISONIK-specific files are uploaded into the BPS with the VISOTOOL Editor.

What happens next?

The following pages provide detailed information on each stage.

4.6 Entering the location text in the RXT10

Purpose of the location text in the RXT10

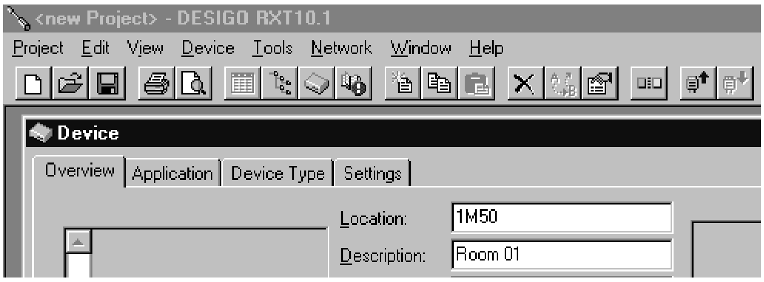
When integrating DESIGO RXC into VISONIK, the “Location” input field in the RXT10 is used for additional VISONIK-specific information about the room number, master/slave controller and RXC controller number. This information is referred to as the “location text”. It can be entered in the following places, depending on the activity being carried out:

– In the “Device” menu, when adding a new device
– In the “List View” menu when creating new devices by copying and modifying identical applications or when editing the data at a later stage.

The information entered is needed by the conversion tool (a component of the RXT10 tool) to create the relevant VISONIK data structures.

Input under “Add devices”

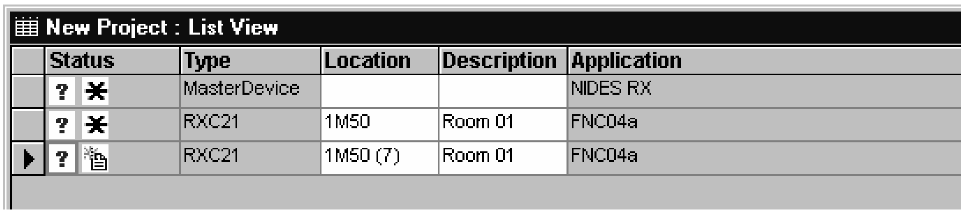
The screenshot below shows the “Overview” tab after selection of the “Device” / “Add” menu.
The location text “1M50” for the new devices has been entered in the “Location” field:



8338Z26E

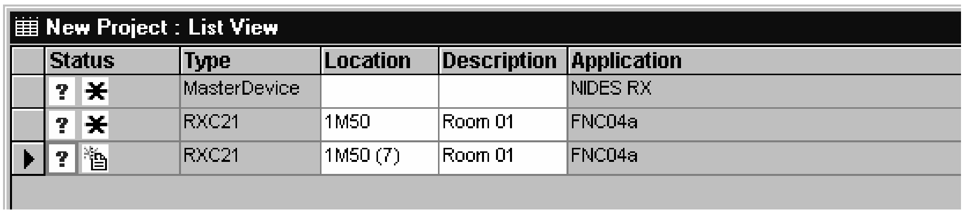
Input under “Copy devices”

The screenshot below shows the situation in the “List View” after copying and adding the newly created device using the options “Device” / “Copy”, “Device” / “Add”.



8338O01E

The location text and room number were then edited in the relevant fields to 10M111/Room 10:



8338O02E

Modifications: IMPORTANT NOTE

The RxNIBPS.COL file is created whenever the project is saved. In the event of changes, such as the later re-allocation of rooms, the changes must also be edited in the RXT10 and a new RxNIBPS.COL file must be generated by saving the new data.

4.6.1 Content and format of the location text

Contents

The location text must always include the following information:

- Number of the room in which the controller is located
- Information as to whether the controller in that room is a master or slave (HVAC)
- VISONIK RXC device number (identification of the device in the VISONIK system)

Note:
The location text does not necessarily correspond to the building-based addresses. These can be entered later with the appropriate text in the TEC data point.

Format

The format of the location text for the integration of RXC controllers into VISONIK is as follows:

rrMccc or **rrSccc**

Character	Description	Value range	Note
rr	Room number	1 ... 99	1 or 2 digits from the VISONIK TEC address range (limited here to 1 … 99)
M	Master controller		There is always 1 HVAC master controller in the room
S	Slave controller		Apart from the master controller, all the RXC controllers in the room are slave controllers (max. 9 slaves allowable)
ccc	RXC controller number	33 ...126	2 or 3 digits from the VISONIK TEC address range

Format for third-party devices

The format of the location text for the integration of third-party devices is:

rr Room number 1 ... 99
X Identifies devices as a third-party device
ccc Device number 128 ... 255
Example: 47X163

N.B.: No leading zero!

Do enter leading zeros for the location text in the RXT10, otherwise the RX master (CFE application) will not run.

Example:
Enter **“1M50”** – not “01M050”.

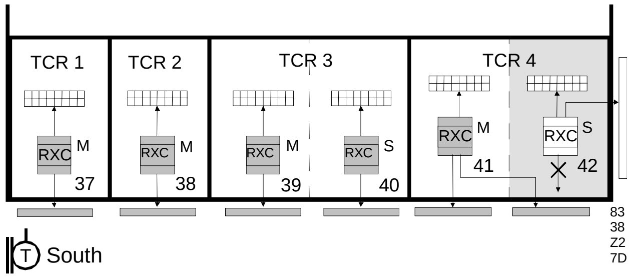
See also the location text entries for the example on the next page.

Note on M and S

All RXC applications have **HVAC** functions. Some

– have **no** lighting and blind functions, while others
– **do** have lighting and blind functions

The allocation of M or S in the location text field is determined only by the HVAC functions of the application.



Location text

The location text for this example is as follows:

New Project : List View

Status	Type	Location	Description	Application
? *	MasterDevice			NIDES RX
? *	RXC3041	1M37	Room 01	INT04a
? *	RXC3041	2M38	Room 02	INT04a
? *	RXC3041	3M39	Room 03	INT04a
? *	RXC3041	3S40	Room 03	INT04a
? *	RXC3041	4M41	Room 04	INT04a
? *	RXC3041	4S42	Room 04	INT04a

8338O03E

Note

The input format and the value ranges are checked by the RXT10 tool. However, there is no check when the data is entered; checking only takes place when you save the project.

Devices which do not have bindings to the NIDES.RX will not be checked, and the associated location text can be defined freely.

The entry in the "Location" input field must comply with the VISONIK syntax rules. The information in the "Location" field serves to identify the relationship between interconnected devices (e.g. a QAX51 room unit connected to an RXC30.1 room controller).

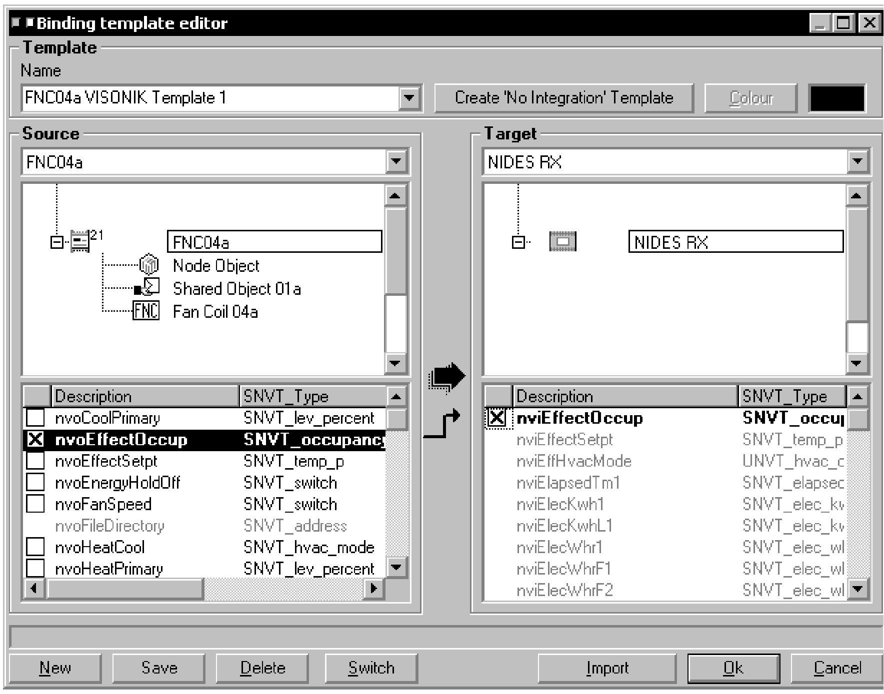
4.7 Selecting the NIDES binding template

Template selection

In the RXT10, select NIDES binding template VISONIK 1 or VISONIK 2.

User-defined templates

To create user-defined templates, use the Binding Template Editor. The screenshot shown is valid for the direction “Device to NIDES.R”:



X”.

Clicking the “Switch” button allows you to change direction (network variables from the NIDES.RX to the device).

Notes

For general information on using the Binding Template Editor, refer to the manual for the RXT10 commissioning and service tool.

User-defined templates can be created simply by basing them on the standard templates and adding or removing individual bindings.

There are no standard templates for the integration of third-party devices.

The general procedure for integrating network variables is described in 4.11, Integration of single parameters.

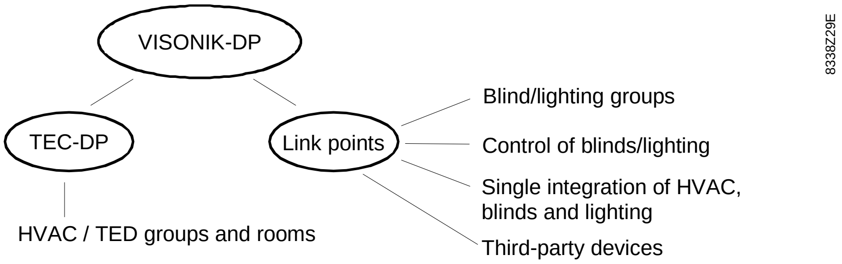
4.8 VISONIK data structures for integration

Introduction

The following engineering steps deal with the mapping of the RXC controllers and their functions to VISONIK data points. This pave gives an overview of the VISONIK data structures used in this process.

Overall classification

The diagram shows the overall classification:



Data points used

The following table shows which data points are used to represent the different categories of plant:

Plant category	Data points	
HVAC functions	TCG	Group point for rooms
	VIP	Point for group setpoint reset
	TCR	Room point, representing controllers in the room
	TED	Energy demand point
Blind function	@AOn functions	Blind group in RXC controller with blind-control
	@AOx'1.0.4	Control of blinds 1 … 4
Lighting functions	@AOn functions	Lighting group in RXC controller with lighting-control
	@AOx'5..8	Control of lighting zones 1 4
Single parameters	@AOx'y	Device outputs
	@AIx'y	Device inputs
		y in the range 11…210

Reserved address ranges

The following address ranges are reserved for RXC integration:

VISONIK DP	Address range
TCG	1 … 255
VIP	1 ... 255
TCR	1 … 99
TED	1 … 255

VISONIK DP	Address range
AOn	n = 1.. 200
AOx ' y AI x ' y	y = 1 .. 210 x = 33 .. 126 for RXC x = 128 .. 255 for 3rd party

Automatic data point generation

In the reserved address ranges, the NIDES.RX-CFE software automatically generates the VISONIK data points required for integration. They are generated on the basis of the configuration data (TSK111 in file RxNiBps.col and TRSK 112 in the GroupConfig.col file). For this reason, special care is required when creating additional data points.

Under no circumstances should additional TEC points (TCR, TCG and TED) be generated.

In the case of VIP points, note that for each @TCGm a @VIPm (with the same address) is generated automatically for the setpoint shift function.

4.9 Defining groups

Introduction

The groupings are defined in file GroupConfig.COL by editing TSK112 manually (assuming it exists; if not, it must be created). In this task the “RX coupling” application later creates the VISONIK data structures automatically.

Default TSK112

The print-out below shows the default TSK112 with details of the format for each of the groups. For editing and saving the GroupCOnfig.COL file, we recommend use of the VISOTool, because in the BPS the comments are removed when the file is saved. The arrows in the left margin point to the first input line of each "group category" (HVAC groups, lighting groups etc.)

```
.TSK112
 1 -- CFE-NIDES.RX, V2.0 : Group Configuration
 100 NewG:=1
 110 END

 ;990 HVAC groups *****************************************************
 ;991 Format: DATA group number,room number,room number,room number,-9
 ;992         DATA group number,room number,room number,room number,-9
 ;993         DATA -9
 ;994 Ranges: group number --> 1..200
 ;995         room number --> 1..99
⇒ ;996 *****************************************************************
 1000 DATA -9

 ;1990 TED groups (Energy demand) *************************************
 ;1991 Format: DATA group number,EFORM,regulator number,regulator number,-9
 ;1992         DATA group number,EFORM,regulator number,regulator number,-9
 ;1993         DATA -9
 ;1994 Ranges: group number --> 1..200
 ;1995         EFORM --> 0..31
 ;1996         regulator number --> 33..126
⇒ ;1997 ****************************************************************
 2000 DATA -9

 ;2990 Blind groups ***************************************************
 ;2991 Format: DATA group number,priority,regulator number,regulator number,-9
 ;2992         DATA group number,priority,regulator number,regulator number,-9
 ;2993         DATA -9
 ;2994 Ranges: group number --> 1..100
 ;2995         priority --> 1..100
 ;2996         regulator number --> 33..126
⇒ ;2997 ****************************************************************
 3000 DATA -9

 ;3990 Light groups ***************************************************
 ;3991 Format: DATA Group number,priority,regulator number,regulator number,-9
 ;3992         DATA Group number,priority,regulator number,regulator number,-9
 ;3993         DATA -9
 ;3994 Ranges: Group number --> 101..200
 ;3995         priority --> 1..100
 ;3996         regulator number --> 33..126
⇒ ;3997 ****************************************************************

4000 DATA -9
;4996 TTY interface to NIDES.RX **************************************
 ;4997 Format: 1 --> TTY1, 2 --> TTY2 or -9 --> TTY1
⇒ ;4998 ****************************************************************
 5000 DATA -9
EXIT
```

4.9.1 Defining groups in GroupConfig.COL

Edit the GroupConfig.COL file as indicated in the table below:

Element	**Entries**
DATA -9	Enter a line for each group, using the format indicated above the input lines. Each line must start with DATA and end with the terminator " –9 ". DATA –9 at the end of the inputs terminates the definition of the groups.
HVAC groups	HVAC groups: Group numbers 1 .. 200, and max. 20 rooms per group *Important:* A room must not be assigned to more than one HVAC group.
TED groups	TED groups: Group numbers 1..200, EFORM parameters in TED (see Part 3 of the documentation on FLN Points).
Blind groups	Blind groups: Group numbers 1..100, with logical priorities assigned in accordance with the plant functions.
Light groups	Lighting groups: Group numbers 101.. 200, priorities as above
TTY interface to NIDES.RX	Enter the TTY interface to the NIDES.RX.

Unedited TSK112

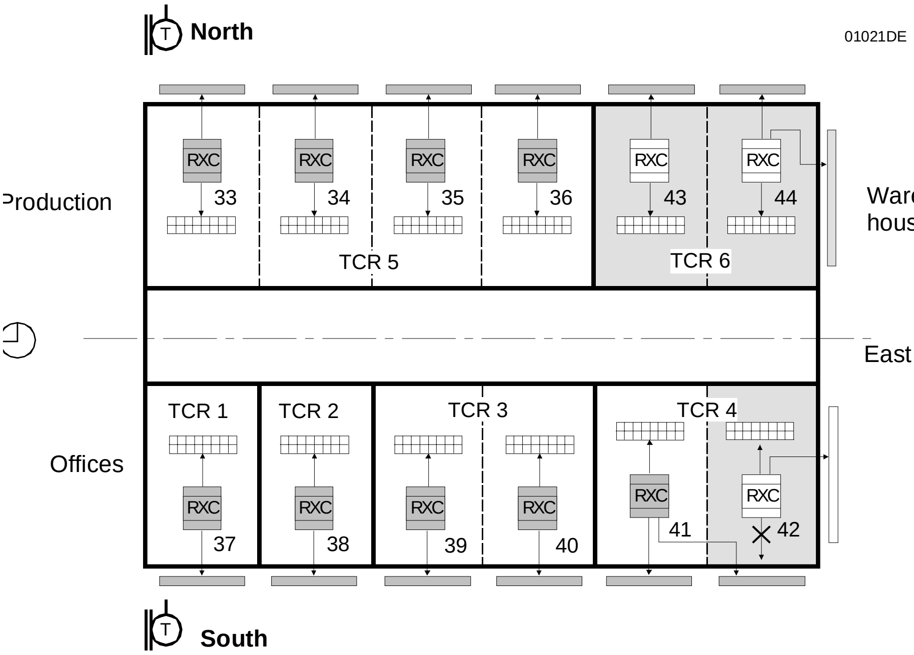
If GroupConfig.COL is not edited

– there will be no groups (HVAC, TED, blinds, lighting) and
– TTY1 will be assigned to the NIDES.RX as the default.

4.10 Example with HVAC, blind and lighting groups

System example

The following is an example of a floor plan, incorporating DESIGO RXC controllers for the room management system.



Groups in the system example

The system example incorporates the following diagrams:

Groups	Description
HVAC	Production area, offices (HVAC zones North and South) The warehouse is not defined as a group (separate area at constant temperature)
Blinds	Production area, warehouse, North façade, South façade, East façade and entire floor
Lighting	Production area, warehouse, offices and entire floor

Notes on the entries in TSK112

The edited version of TSK112 for this system example is shown on the next page. The following details relate to the edited file:

Groups	Description
HVAC	TSK112 does not include TCR 6 (RX43/44), as it is not a group.
TED	The TED groups are “independent” of the HVAC groups. Four groups have been defined here, on the assumption that the North and South zones each have 2 energy distribution systems with: Heating, water (EFORM = 0) and Cooling, water (EFORM = 1).
Blinds	Group numbers, from 1.. 100 For each group, a priority is entered, which should reflect the importance of the group’s function within the overall system. For example, the warehouse has been assigned a lower priority than the floor as a whole.
Lighting	Group numbers, from 101.. 200 Priorities assigned according to the same principles as for blinds.
TTY	Not edited. ⇒ TTY1 is assigned to the NIDES.RX (default).

4.10.1 Edited GroupConfig.COL for the example

Entries in GroupConfig.COL

The print-out below shows the GroupConfig.COL file, after editing for the system example described above. The edited text is shown in bold. Refer to the previous page for notes on the entries.

```
.TSK112

    1 -- [1-08-99/V1.8/fl] RX-Master: Konfigurations-

  990 -- HVAC group ******************************************************
  991 ; -- Format: DATA group number,room number,room number,room number,-9
  992 ; --         DATA group number,room number,room number,room number,-9
  993 ; --         DATA -9
  994 ; -- Range: group number --> 1..200
  995 ; --        room number --> 1..99
  996 ; -- ****************************************************************
 1000 DATA 1,5,-9
 1010 DATA 2,1,2,3,4,-9
 1020 DATA -9

 1990 -- TED group (Energy demand)*********************************************
 1991 ; -- Format: DATA group number,EFORM,regulator number,regulator number,-9
 1992 ; --         DATA group number,EFORM,regulator number,regulator number,-9
 1993 ; --         DATA -9
 1994 ; -- Range: group number --> 1..200
 1995 ; --        EFORM --> 0..31
 1996 ; --        regulator number --> 33..126
 1997 ; -- ********************************************************************
 2000 DATA 1,0,33,34,35,36,-9
 2010 DATA 2,1,33,34,35,36,-9
 2020 DATA 3,0,37,38,39,40,41,42,-9
 2030 DATA 4,1,37,38,39,40,41,42,-9
 2040 DATA -9

 2990 -- Blind group ************************************************************
 2991 ; -- Format: DATA group number,priority,regulator number,regulator number,-9
 2992 ; --         DATA group number,priority,regulator number,regulator number,-9
 2993 ; --         DATA -9
 2994 ; -- Range: group number --> 1..100
 2995 ; --        priority --> 1..100
 2996 ; --        regulator number --> 33..126
 2997 ; -- ***********************************************************************
 3000 DATA 1,5,33,34,35,36,-9
 3010 DATA 2,6,43,44,-9
 3020 DATA 3,4,33,34,35,36,43,44,-9
 3030 DATA 4,2,37,38,39,40,41,42,-9
 3040 DATA 5,3,42,44,-9
 3050 DATA 6,1,33,34,35,36,43,44,37,38,39,40,41,42,-9
 3060 DATA -9

 3990 -- Light group ************************************************************
 3991 ; -- Format: DATA Group number,priority,regulator number,regulator number,-9
 3992 ; --         DATA Group number,priority,regulator number,regulator number,-9
 3993 ; --         DATA -9
 3994 ; -- Range: Group number --> 101..200
 3995 ; --        priority --> 1..100
 3996 ; --        regulator number --> 33..126
 3997 ; -- ***********************************************************************
 4000 DATA 101,3,33,34,35,36,-9
 4010 DATA 102,4,43,44,-9
 4030 DATA 103,2,37,38,39,40,41,42,-9
 4000 DATA 104,1,33,34,35,36,43,44,37,38,39,40,41,42,-9
 4040 DATA -9

 4996 -- TTY interface to NIDES.RX **********************
 4997 ; -- Format: 1 --> TTY1, 2 --> TTY2 oder -9 --> TTY1
 4998 ; -- ***********************************************
 5000 DATA -9

EXIT
```

4.10.2 Lighting and blind group priorities

The following rules apply to the priorities:

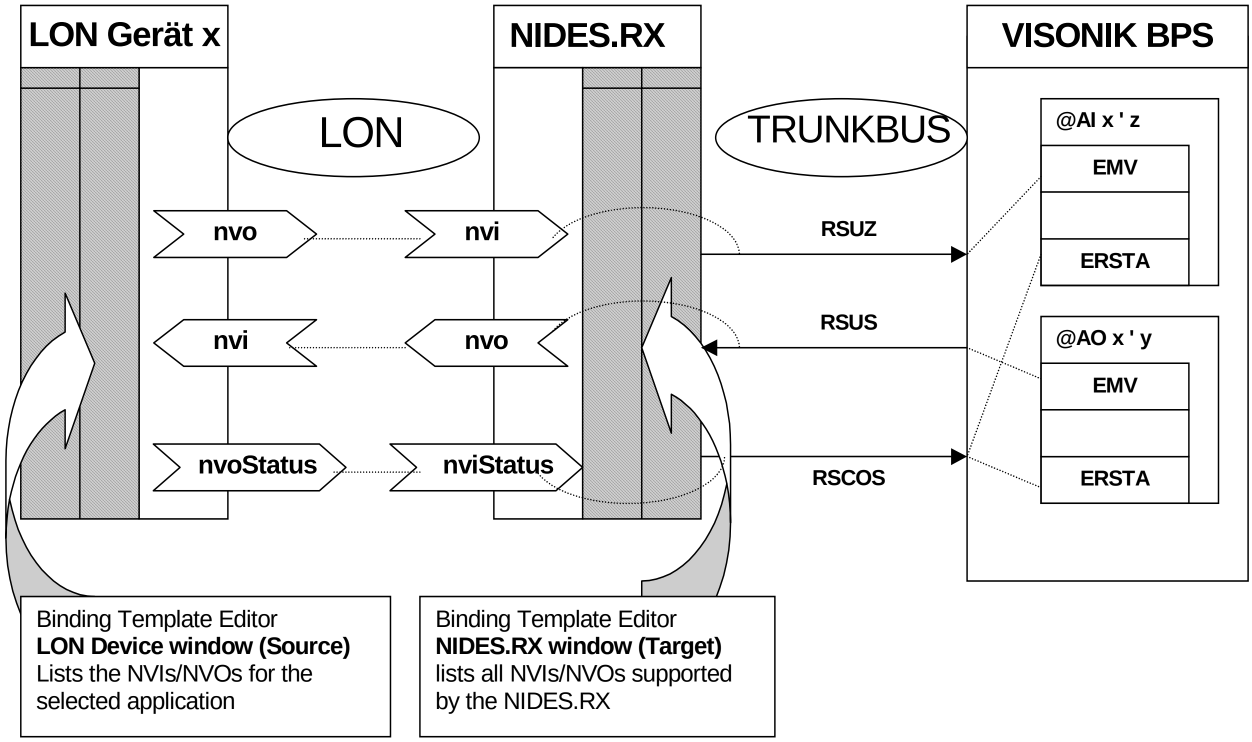
- Each group of blinds and each lighting group must be assigned a priority level between 1 and 100, where 1 is the highest.
- No priority level may be assigned more than once (this applies to lighting and blinds separately).
- The priorities assigned are relevant only if the group concerned is configured in “Remote” mode.
- In "Remote" mode, if two groups with the same priority act on the same controllers, the last command always applies.

4.11 Integration of single parameters

The following diagram applies to both the additional network variables of standard RXC applications and to the network variables of third-party applications.

The term “single parameter integration” refers to the mapping of an individual NV element to the Event Main Value (EVM) of a link point (AI or AO) without any special additional function.

The diagram below shows the principle of this mapping process.



(The term “LON device” refers to both RXC and third-party device.)

Mapping rule for single parameters

An input **(nvi)** of the NIDES.RX is mapped to the EMV of an **AI** point.

An output **(nvo)** of the NIDES.RX is mapped to the EMV of an **AO** point.
The link point address consists of two parts: **Group address'Element address**
The **group address** of the link point is **Device number x**.
The **Element address** of the link point identifies the input **(nvi)** or output **(nvo)** on the NIDES.RX side. The allocation of inputs/outputs to the element addresses is fixed, and documented in Tables 4.10.1 and 4.10.2.

Exception: Mapping the Object Status

The NIDES.RX input **nviStatus (Object Status)** is mapped to parameter ERSTA for all VISONIK points that belong to the LON device concerned (cf. Mapping rule in Section .
The Object Status must be integrated explicitly for each third-party device.

Restrictions affecting single parameter integration

Network variables can be integrated as single parameters, provided that:

- the RXT10 tool knows the application concerned
- the allocation from input (nvi) or output (nvo) on the NIDES.RX side to the link point element address is defined. **Caution:** This only applies to some of the NVs shown in the NIDES.RX window (target). NVs not documented in Tables 4.10.1 and 4.10.2 are excluded from single parameter integration.

The network variables of RXC applications for which an integration function is proposed (see Sections 3.2.3 to 3.2.28) are excluded from single parameter integration.

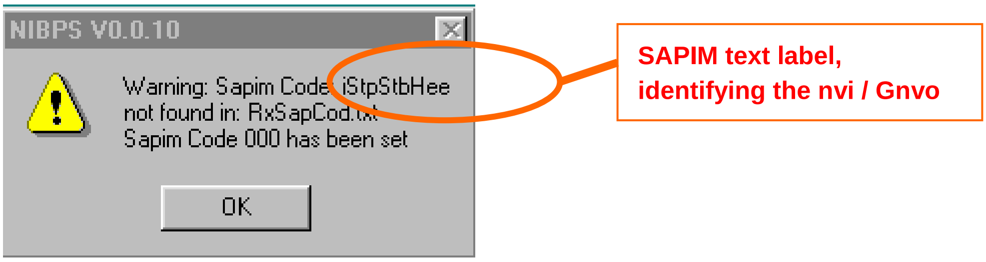
For third-party devices (device numbers equal to or greater than 128), the integration functions are disabled. This means that all the NVs documented in Tables 4.10.1 and 4.10.2 can be integrated as single parameters.

Engineering the single parameters

To integrate a single parameter, an appropriate binding must be created using the Binding Template Editor of the RXT10 tool. In this process, only the input/output on the NIDES.RX side is relevant for integration purposes. In conjunction with Device No. x, it defines the associated link point.

The source for the input/ouput on the NIDES.RX side can be selected freely, provided it is of the same SNVT type. When the project is saved, the RXT10 tool generates an entry for each binding of this kind in the configuration file RxNiBps.col for the BPS.

If the nvi/nvo has not been allocated to a link point, an error message will appear:



Caution: Before you continue to work, please delete this binding or change to another one. During the save operation, the above pop-up must not appear.
Otherwise the stability of the application will suffer seriously.

The NIDES.RX-CFE software automatically generates the link point associated with the binding, on the basis of the entry in file RxNiBps.col.

Text commentary for link points

When it is generated, every link point is assigned a text number TXI2 (Information Text, Part 2) in the range Ig1001 to Ig1408 of the global VISONIK text catalog. This range includes the names of all the network variables that can be integrated. The name of the relevant NV is therefore shown for each integrated single parameter in the BPS and DCS reports.

Example: If the NV Application Mode is integrated, the following display appears in a `DIR @AO` on the BPS.

```
@AO128'26        ; --      nviApplicMode
```

Note that here, the NV is qualified as an input (on the device side).
The direction of data flow is unambiguously defined by the type of link point.

Text download

The text items required are normally loaded from the DCS to the BPS via the DCS command **LoTx**. This presupposes that all the link points of the project have already been generated on the DCS and that they have all been assigned the correct text numbers. In other words, the text must not be loaded until the process image has been created in the DCS (Command **SaPa**). Refer to the section on “Commissioning the BPS” in the Engineering Guide.

4.11.1 Mapping table for NIDES.RX outputs

The **NV Name** and **SNVT_Type** columns contain the names of the NIDES.RX outputs in the order in which they appear in the NIDES.RX window of the Binding Template Editor. The switch must be set so that the NIDES.RX window

(Target) is to the left of the connecting arrow. ⇨

Outputs in the NIDES.RX window of the Binding Template Editor cannot be integrated unless they are documented here.

In the case of structured NVs, a permanently defined selection of elements is mapped. These are shown in the ***NV Element*** column (not visible in the NIDES.RX window). The value range, unit and semantics are defined by the SNVT type and documented here.

The ***Link point*** column shows the associated address. The link point type is always AO, and the output value is always the EMV of the link point.

The **Integration function** column contains a reference to the relevant subsection of Section 3. NVs with this type of reference can only be integrated with third-party devices. NVs with the reference **AppLibV2** are available when use is made of Application Library V2, and can be integrated as single value parameters.

NIDES.RX				**BPS**	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
ncoLuxThrsOff_1	SNVT_lux		←	@AO x ' 112	
ncoLuxThrsOff_2	SNVT_lux		←	@AO x ' 113	
ncoOAMinPos	SNVT_lev_percent		←	@AO x ' 68	
ncoSetpoints	SNVT_temp_setpt	occupied_cool	←	@AO x ' 62	3.2.11
		occupied_heat	←	@AO x ' 65	3.2.11
		standby_cool	←	@AO x ' 63	3.2.11
		standby_heat	←	@AO x ' 66	3.2.11
		unoccupied_cool	←	@AO x ' 64	3.2.11
		unoccupied_heat	←	@AO x ' 67	3.2.11
ncoVolSetpt	UNVT_vol_setpt	MaxDischAirFlow	←	@AO x ' 188	AppLibV2
		MaxFlow	←	@AO x ' 182	
		MaxFlowHeat	←	@AO x ' 183	
		MinFlow	←	@AO x ' 184	
		MinFlowEco	←	@AO x ' 185	
		MinFlowHeat	←	@AO x ' 186	
		MinFlowStby	←	@AO x ' 187	

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nvoAirflowSetpt	SNVT_flow		←	@AO x ' 19	
nvoApplicMode	SNVT_hvac_mode		←	@AO x ' 26	3.2.4
nvoAuxHeatEnable	SNVT_switch	Value	←	@AO x ' 30	3.2.23
nvoCount_1	SNVT_count		←	@AO x ' 99	
nvoCount_2	SNVT_count		←	@AO x ' 100	
nvoCtrl_Q14	SNVT_switch	State	←	@AO x ' 141	
nvoCtrl_Q24	SNVT_switch	State	←	@AO x ' 143	
nvoCtrl_Q34	SNVT_switch	State	←	@AO x ' 145	
nvoCtrl_Y1	SNVT_switch	State	←	@AO x ' 133	
nvoCtrl_Y2	SNVT_switch	State	←	@AO x ' 135	
nvoCtrl_Y3	SNVT_switch	State	←	@AO x ' 137	
nvoCtrl_Y4	SNVT_switch	State	←	@AO x ' 139	
nvoCtrl_Y5	SNVT_switch	State	←	@AO x ' 20	
nvoCtrl_Y6	SNVT_switch	State	←	@AO x ' 36	
nvoEconEnable	SNVT_switch	State	←	@AO x ' 33	
nvoEmergOverride	SNVT_hvac_emerg		←	@AO x ' 42	
nvoEnergyHoldOff	SNVT_switch	State	←	@AO x ' 35	
nvoExMod_1_Q14	SNVT_switch	State	←	@AO x ' 147	
nvoExMod_1_Q24	SNVT_switch	State	←	@AO x ' 149	
nvoExMod_1_Q34	SNVT_switch	State	←	@AO x ' 151	
nvoExMod_1_Q44	SNVT_switch	State	←	@AO x ' 153	
nvoExMod_2_Q14	SNVT_switch	State	←	@AO x ' 155	
nvoExMod_2_Q24	SNVT_switch	State	←	@AO x ' 157	
nvoExMod_2_Q34	SNVT_switch	State	←	@AO x ' 159	
nvoExMod_2_Q44	SNVT_switch	State	←	@AO x ' 161	
nvoFanSpeedCmd	SNVT_switch	State	←	@AO x ' 29	
nvoFhMode	UNVT_fh_mode	State	←	@AO x ' 189	AppLibV2
		workmode	←	@AO x ' 190	
nvoFlow_1	SNVT_flow		←	@AO x ' 94	
nvoFlow_2	SNVT_flow		←	@AO x ' 95	

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nvoHeatCool	SNVT_hvac_mode		←	@AO x ' 27	
nvoHeatSrcTemp	SNVT_temp_p		←	@AO x ' 53	
nvoHvacMode	SNVT_hvac_mode		←	@AO x ' 96	
nvoLevCont_1	SNVT_lev_cont		←	@AO x ' 101	
nvoLevCont_2	SNVT_lev_cont		←	@AO x ' 102	
nvoLevCont_3	SNVT_lev_cont		←	@AO x ' 103	
nvoLevCont_4	SNVT_lev_cont		←	@AO x ' 104	
nvoLevDisc_1	SNVT_lev_disc		←	@AO x ' 105	
nvoLevDisc_2	SNVT_lev_disc		←	@AO x ' 106	
nvoLevPercent_1	SNVT_lev_percent		←	@AO x ' 89	
nvoLevPercent_2	SNVT_lev_percent		←	@AO x ' 90	
nvoLevPercent_3	SNVT_lev_percent		←	@AO x ' 91	
nvoLightCmd_1	SNVT_switch	Value	←	@AO x ' 75	3.2.27
nvoLightCmd_2	SNVT_switch	Value	←	@AO x ' 79	3.2.27
nvoLightCmd_3	SNVT_switch	Value	←	@AO x ' 83	3.2.28
nvoLightCmd_4	SNVT_switch	Value	←	@AO x ' 87	3.2.28
nvoLightLuxLevel	SNVT_lux		←	@AO x ' 111	
nvoLightOverr_1	SNVT_switch	State	←	@AO x ' 108	3.2.27
nvoLightOverr_2	SNVT_switch	State	←	@AO x ' 110	3.2.27
nvoLightOverr_3	SNVT_switch	State	←	@AO x ' 195	
nvoLightOverr_4	SNVT_switch	State	←	@AO x ' 196	
nvoLux	SNVT_lux		←	@AO x ' 98	
nvoMmiEnable	SNVT_switch	State	←	@AO x ' 191	AppLibV2
		Value	←	@AO x ' 192	
nvoMute	SNVT_switch	State	←	@AO x ' 193	AppLibV2
		Value	←	@AO x ' 194	
nvoOccManCmd	SNVT_occupancy		←	@AO x ' 24	
nvoOccSchedule	SNVT_tod_event	Current_state	←	@AO x ' 21	3.2.3
nvoOccSensor	SNVT_occupancy		←	@AO x ' 25	
nvoOccupancy_1	SNVT_occupancy		←	@AO x ' 92	
nvoOccupancy_2	SNVT_occupancy		←	@AO x ' 93	
nvoOutdoorDewPt	SNVT_temp_p		←	@AO x ' 48	

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nvoOutdoorTemp	SNVT_temp_p		←	@AO x ' 44	
nvoPpm	SNVT_ppm		←	@AO x ' 97	
nvoPressP_1	SNVT_press_p		←	@AO x ' 162	
nvoSblndCmd_1	SNVT_setting	function	←	@AO x ' 114	3.2.25
nvoSblndCmd_2	SNVT_setting	function	←	@AO x ' 117	3.2.25
nvoSblndCmd_3	SNVT_setting	function	←	@AO x ' 120	3.2.26
nvoSblndCmd_4	SNVT_setting	function	←	@AO x ' 123	3.2.26
nvoSblndOverr_1	SNVT_setting	function	←	@AO x ' 126	3.2.25
nvoSblndOverr_2	SNVT_setting	function	←	@AO x ' 129	3.2.25
nvoSblndOverr_3	SNVT_setting	function	←	@AO x ' 197	
nvoSblndOverr_4	SNVT_setting	function	←	@AO x ' 198	
nvoScene_1	SNVT_scene	function	←	@AO x ' 132	
		scene_number	←	@AO x ' 134	
nvoSceneCfg_1	SNVT_scene_cfg	delay_time	←	@AO x ' 146	
		fade_time	←	@AO x ' 144	
		function	←	@AO x ' 136	
		priority	←	@AO x ' 148	
		Rotation	←	@AO x ' 142	
		scene_number	←	@AO x ' 138	
		Setting	←	@AO x ' 140	
nvoSetptOffset	SNVT_temp_p		←	@AO x ' 12	
nvoSetptShift	SNVT_temp_setpt	occupied_cool	←	@AO x ' 13	3.2.9
		occupied_heat	←	@AO x ' 16	3.2.9
nvoSetting_1	SNVT_setting	function	←	@AO x ' 73	
		Rotation	←	@AO x ' 76	
		Setting	←	@AO x ' 74	
nvoSetting_2	SNVT_setting	function	←	@AO x ' 77	
		Rotation	←	@AO x ' 80	
		Setting	←	@AO x ' 78	
nvoSetting_3	SNVT_setting	function	←	@AO x ' 81	
		Rotation	←	@AO x ' 84	
		Setting	←	@AO x ' 82	

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nvoSetting_4	SNVT_setting	function	←	@AO x ' 85	
		Rotation	←	@AO x ' 88	
		Setting	←	@AO x ' 86	
nvoSourceTemp	SNVT_temp_p		←	@AO x ' 43	
nvoSpaceCO2	SNVT_ppm		←	@AO x ' 46	
nvoSpaceDewPt	SNVT_temp_p		←	@AO x ' 47	
nvoSpaceRH	SNVT_temp_p		←	@AO x ' 45	
nvoSpaceTemp	SNVT_temp_p		←	@AO x ' 11	3.2.21
nvoSpeed_1	SNVT_speed		←	@AO x ' 150	
nvoSpeed_2	SNVT_speed		←	@AO x ' 152	
nvoSwitch_1	SNVT_switch	State	←	@AO x ' 40	
		Value	←	@AO x ' 39	
nvoSwitch_2	SNVT_switch	State	←	@AO x ' 49	
		Value	←	@AO x ' 41	
nvoSwitch_3	SNVT_switch	State	←	@AO x ' 55	
		Value	←	@AO x ' 54	
nvoSwitch_4	SNVT_switch	State	←	@AO x ' 57	
		Value	←	@AO x ' 56	
nvoSwitch_5	SNVT_switch	State	←	@AO x ' 59	
		Value	←	@AO x ' 58	
nvoSwitch_6	SNVT_switch	State	←	@AO x ' 61	
		Value	←	@AO x ' 60	
nvoSwitch_7	SNVT_switch	State	←	@AO x ' 70	
		Value	←	@AO x ' 69	
nvoSwitch_8	SNVT_switch	State	←	@AO x ' 72	
		Value	←	@AO x ' 71	
nvoTempP_1	SNVT_temp_p		←	@AO x ' 31	
nvoTempP_2	SNVT_temp_p		←	@AO x ' 32	
nvoTempP_3	SNVT_temp_p		←	@AO x ' 34	
nvoTempP_4	SNVT_temp_p		←	@AO x ' 37	
nvoTempP_5	SNVT_temp_p		←	@AO x ' 38	

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nvoTimeSec_1	SNVT_time_sec		←	@AO x ' 154	
nvoTimeSec_2	SNVT_time_sec		←	@AO x ' 156	
nvoTimeSec_3	SNVT_time_sec		←	@AO x ' 158	
nvoTimeSec_4	SNVT_time_sec		←	@AO x ' 160	
nvoTimeStamp_1	SNVT_time_stamp	day	←	@AO x ' 165	
		hour	←	@AO x '166	
		minute	←	@AO x ' 167	
		month	←	@AO x ' 164	
		second	←	@AO x ' 168	
		year	←	@AO x ' 163	
nvoTodEvent_1	SNVT_tod_event	Current_state	←	@AO x ' 169	
		next_state	←	@AO x ' 170	
		time_to_next_state	←	@AO x ' 171	
nvoTodEvent_2	SNVT_tod_event	Current_state	←	@AO x ' 172	
		next_state	←	@AO x ' 173	
		time_to_next_state	←	@AO x ' 174	
nvoTodEvent_3	SNVT_tod_event	Current_state	←	@AO x ' 175	
		next_state	←	@AO x ' 176	
		time_to_next_state	←	@AO x ' 177	
nvoTodEvent_4	SNVT_tod_event	Current_state	←	@AO x ' 178	
		next_state	←	@AO x ' 179	
		time_to_next_state	←	@AO x ' 180	
nvoUseSchedule	SNVT_tod_event	Current_state	←	@AO x ' 50	3.2.3
nvoValveOverride	SNVT_hvac_overid	State	←	@AO x ' 181	

4.11.2 Mapping table for NIDES.RX inputs

The ***NV Name*** and ***SNVT_Type*** columns contain the names of the NIDES.RX inputs in the order in which they appear in the NIDES.RX window of the Binding Template Editor. The switch must be set so that the NIDES.RX window

(Target) is to the right of the connecting arrow.

Inputs in the NIDES.RX window of the Binding Template Editor cannot be integrated unless they are documented here.

In the case of structured NVs, a permanently defined selection of elements is mapped. These are shown in the ***NV Element*** column (not visible in the NIDES.RX window). The value range, unit and semantics are defined by the SNVT type and documented here.

The ***Link point*** column shows the associated address. The Link Point Type is always *AI* and the output value is always the EMV of the lnk point.

Sole exception: nviStatus (Object Status) is mapped to the ERSTA (see 3.2.2).
In the case of third-party devices, the nviStatus must be integrated explicitly.

The ***Integration function*** column contains a reference to the relevant subsection of Section 3.

- NVs with this type of reference can only be integrated with third-party devices.
- NVs with the reference ***AppLibV2*** are available when use is made of Application Library V2, and can be integrated as single value parameters.
- NVs with the reference **AppLibV1** are not available in Application Library V2. This applies to the NVs **nviSetptEffect** and **nviEffectOccup** which have been replaced in V2 by **nviEffHvacMode**.

NIDES.RX				**BPS**	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nviAbsHumid1	SNVT_abs_humid		→	@AI x ' 141	AppLibV2
nviAirDamperPos	SNVT_lev_percent		→	@AI x '85	3.2.17
nviAirDamperPos2	SNVT_lev_percent		→	@AI x '14	3.2.17
nviAirflow	SNVT_flow		→	@AI x '73	3.2.16
nviAirflow2	SNVT_flow		→	@AI x '15	3.2.16
nviAirFlow2Fh	SNVT_flow		→	@AI x '125	AppLibV2
nviAirFlow2Tot	SNVT_flow		→	@AI x '127	AppLibV2
nviAirFlow2TotDem	SNVT_flow		→	@AI x '126	AppLibV2
nviAirflowHeat	SNVT_flow		→	@AI x '87	3.2.16
nviAmp1	SNVT_amp		→	@AI x '142	AppLibV2
nviAmpAc1	SNVT_amp_ac		→	@AI x '143	AppLibV2
nviAmpF1	SNVT_amp_f		→	@AI x '144	AppLibV2

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nviCoolPrimary	SNVT_lev_percent		→	@AI x '65	3.2.12
nviCoolSecondary	SNVT_lev_percent		→	@AI x '12	3.2.12
nviCount1	SNVT_count		→	@AI x '79	
nviCtrl_B1	SNVT_temp_p		→	@AI x '153	
nviCtrl_D1	SNVT_switch		→	@AI x '155	
nviCtrl_D2	SNVT_switch		→	@AI x '157	
nviCtrl_D3	SNVT_switch		→	@AI x '159	
nviCtrl_D4	SNVT_switch		→	@AI x '161	
nviDischAirTemp	SNVT_temp_p		→	@AI x '60	
nviEffectOccup	SNVT_occupancy		→	@AI x '50	3.2.6 AppLib**V1**
nviEffectOccup	SNVT_occupancy		→	@AI x '51	AppLib**V2**
nviEffectSetpt	SNVT_temp_p		→	@AI x '42	
nviEffectFlowSP	SNVT_flow		→	@AI x '71	
nviEffHvacMode	UNVT_hvac_op_mde	cool_setpt	→	@AI x '74	3.2.8 AppLib**V2**
		heat_setpt	→	@AI x '77	3.2.8 AppLib**V2**
		op_mode	→	@AI x '50	3.2.6 AppLib**V2**
nviEnergyHoldOff	SNVT_switch	State	→	@AI x '70	3.2.18
nviEnthalpy1	SNVT_enthalpy		→	@AI x '145	AppLibV2
nviExhaustFlow	SNVT_flow		→	@AI x '121	AppLibV2
nviExMod_1_D1	SNVT_switch	State	→	@AI x '163	
nviExMod_1_D2	SNVT_switch	State	→	@AI x '165	
nviExMod_1_D3	SNVT_switch	State	→	@AI x '167	
nviExMod_1_D4	SNVT_switch	State	→	@AI x '169	
nviExMod_2_D1	SNVT_switch	State	→	@AI x '171	
nviExMod_2_D2	SNVT_switch	State	→	@AI x '173	
nviExMod_2_D3	SNVT_switch	State	→	@AI x '175	
nviExMod_2_D4	SNVT_switch	State	→	@AI x '177	
nviFaceVelocity	SNVT_speed_mil		→	@AI x '122	AppLibV2
nviFanSpeed	SNVT_switch	State	→	@AI x '59	3.2.13
nviFhMode	UNVT_fh_mode	State	→	@AI x '111	AppLibV2
		workmode	→	@AI x '112	

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nviFlowF1	SNVT_flow_f		→	@AI x '146	AppLibV2
nviFlowP1	SNVT_flow_p		→	@AI x '147	AppLibV2
nviFlowTemp	SNVT_tem_p		→	@AI x '82	
nviFlowSetpoint	SNVT_lev_percent		→	@AI x '72	
nviFreqF1	SNVT_freq_f		→	@AI x '148	AppLibV2
nviHeatAirDmpPos	SNVT_lev_percent		→	@AI x '86	3.2.17
nviHeatCool	SNVT_hvac_mode		→	@AI x '52	
nviHeatPrimary	SNVT_lev_percent		→	@AI x '63	3.2.12 3.2.14
nviHeatSecondary	SNVT_lev_percent		→	@AI x '64	3.2.15
NviHvacMS2	Unsigned (1bit)	master_slave	→	@AI x '180	
	SNVT_temp_p	space_temp	→	@AI x '181	
	SNVT_temp_p	setpt_offset	→	@AI x '182	
	T_Glo_ PPSStatusOpState	room_status	→	@AI x '183	
	T_Glo_Sequence	Setpt_offset_changed	→	@AI x '184	
	Unsigned	room_status_changed	→	@AI x '185	
nviLabAlarm	SNVT_state_64	bit0	→	@AI x '113	AppLibV2
		bit8	→	@AI x '114	
		bit16	→	@AI x '115	
		bit24	→	@AI x '116	
		bit32	→	@AI x '117	
		bit40	→	@AI x '118	
		bit48	→	@AI x '119	
		bit56	→	@AI x '120	
nviLevCont1	SNVT_lev_cont		→	@AI x '178	
nviLevCont2	SNVT_lev_cont		→	@AI x '179	
nviLevPercent12	SNVT_lev_percent		→	@AI x '55	
nviLevPercent13	SNVT_lev_percent		→	@AI x '56	
nviLightCmd_1	SNVT_switch	Value	→	@AI x '97	3.2.27
nviLightCmd_2	SNVT_switch	Value	→	@AI x '101	3.2.27
nviLightCmd_3	SNVT_switch	Value	→	@AI x '105	3.2.28
nviLightCmd_4	SNVT_switch	Value	→	@AI x '109	3.2.28

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nviLoadAbs	SNVT_power		→	@AI x '61	3.2.14 3.2.15
nviLux	SNVT_lux		→	@AI x '78	
nviOADamper	SNVT_lev_percent		→	@AI x '66	
nviOccSensor	SNVT_occupancy		→	@AI x '81	
nviPowerF1	SNVT_power_f		→	@AI x '149	AppLib V2
nviPpm	SNVT_ppm		→	@AI x '76	
nviPumpStatus	SNVT_switch	Value	→	@AI x '83	
nviSashPosition	SNVT_switch	State	→	@AI x '123	AppLibV2
		Value	→	@AI x '124	
nviSblndCmd_1	SNVT_setting	Setting	→	@AI x '130	3.2.25
nviSblndCmd_2	SNVT_setting	Setting	→	@AI x '133	3.2.25
nviSblndCmd_3	SNVT_setting	Setting	→	@AI x '136	3.2.26
nviSblndCmd_4	SNVT_setting	Setting	→	@AI x '139	3.2.26
nviScene1	SNVT_scene	function	→	@AI x '84	
		number	→	@AI x '88	
nviSceneCfg1	SNVT_scene_cfg	delay_time	→	@AI x '94	
		fade_time	→	@AI x '93	
		function	→	@AI x '89	
		number	→	@AI x '90	
		priority	→	@AI x '95	
		Rotation	→	@AI x '92	
		Setting	→	@AI x '91	
nviSetptEffect	SNVT_temp_setp	occupied_cool	→	@AI x '74	3.2.8 AppLib**V1**
		occupied_heat	→	@AI x '77	3.2.8 AppLib**V1**
nviSetptOffset	SNVT_temp_p		→	@AI x '80	
nviSpaceCO2	SNVT_ppm		→	@AI x '13	
nviSpaceDewPt	SNVT_temp_p		→	@AI x '68	
nviSpaceRH	SNVT_lev_percent		→	@AI x '67	
nviSpaceTemp	SNVT_temp_p		→	@AI x '41	3.2.21
nviSpeed1	SNVT_speed		→	@AI x '98	
nviSpeedF1	SNVT_speed_f		→	@AI x '150	AppLibV2
nviStatus	SNVT_obj_status		→		–

NIDES.RX				BPS	
NV Name (Description)	*SNVT_Type*	*NV Element*		*Link point*	*Integration function*
nviSwitch24	SNVT_switch	State	→	@AI x '22	
		Value	→	@AI x '21	
nviSwitch25	SNVT_switch	State	→	@AI x '24	
		Value	→	@AI x '23	
nviTempP10	SNVT_temp_p		→	@AI x '17	
nviTempSensor	SNVT_temp_p		→	@AI x '96	3.2.21
nviTempSensorPPS	SNVT_temp_p		→	@AI x '16	3.2.21
nviTerminalLoad	SNVT_lev_percent		→	@AI x '62	3.2.20
nviUnitStatus	SNVT_hvac_status	in_alarm	→	@AI x '49	***)**
		mode	→	@AI x '43	3.2.19
nviVol1	SNVT_vol		→	@AI x '151	AppLibV2
nviVolF1	SNVT_vol_f		→	@AI x '152	AppLibV2
nviVolt	SNVT_volt		→	@AI x '154	AppLibV2
nviVoltAc1	SNVT_volt_ac		→	@AI x '156	AppLibV2
nviVoltF1	SNVT_volt_f		→	@AI x '158	AppLibV2

***) Value 6W41) for FNC 10 or FNC 12:**
With this binding, a frost alarm for the room can be realized.
If in_alarm=0, @Ai x '49 will show the value 6.
If in_alarm=1, @Ai x '49 will show the value 4.
@Ai x '49.LOL has to be set accordingly.

5 Commissioning

5.1 Overview

Introduction

This section describes the commissioning procedure for the integration of DESIGO RXC into VISONIK. The key points are as follows:

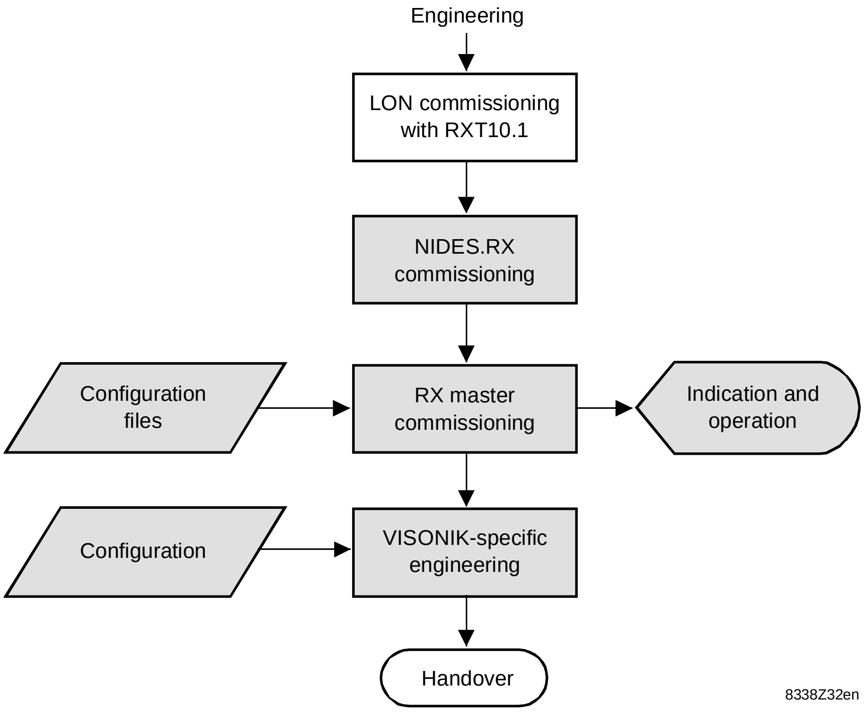
- Commissioning procedure
- Indication and operation
- Procedure for updates and modifications

Contents of this section

5.2 Commissioning procedure

Flow chart

The flow chart below is a simplified illustration of the commissioning procedure for the integration of the DESIGO RXC controllers. The activities and elements specifically related to VISONIK are shown on a shaded background.



Notes on the diagram

The commissioning activities and elements related specifically to VISONIK are as follows:

Activity	Brief description
Commissioning the NIDES	Hardware commissioning as described in data sheet
Commissioning the RX master	Download configuration files / Start RX master / Indication and operation
VISONIK-specific engineering	Same engineering procedure as for TEC (configuration and testing)

A step-by-step commissioning procedure is shown on the next page.

5.3 Commissioning procedure (workflow)

Introduction

This topic deals with commissioning on the VISONIK side. Essentially it consists of the following:

- Commissioning the NIDES.RX
- Commissioning the RX master
- VISONIK-specific engineering

Commissioning the NIDES.RX

To commission the NIDEX.RX, proceed as follows:

Step	Activity
1	Commission the NIDES hardware as described in data sheet CA2N3299E.
2	Connect the NIDES.RX to the RS master *Notes on connection:* – The NIDES.RX and the VISONIK.BPS are connected via V.24 interface, to TTY1 (default) or TTY2, as required. – Serial cable: RS232 cable, data lines 2/3 not crossed, connector: DB9 female / DB25 female.

Commissioning the RX master

The RX master is commissioned using the VISOTOOL Editor. Proceed as follows:

Step	Activity
1	Load the following three files into the BPS: – CFE_V2_x_y.COL (RX master, basic configuration) – GroupConfig.COL (Group definitions) – RxNIBPS.COL (Project-specific data)
2	Run Task 125 (Monitoring task) with "RUN125" *Note:* If GroupConf.COL or or RxNIBPS.COL are not found, "RUN125" will result in an error message with the name of the missing task.

The RX master will now load the Infolist into the NIDES. The process can be followed on the NIDES or the RX master (see "Commissioning procedure (workflow)" on the next page.

VISONIK-specific engineering

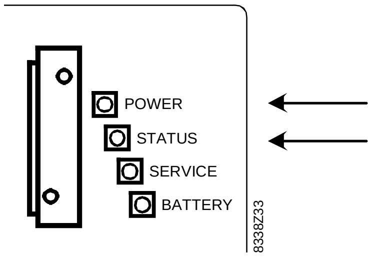
You can then start the VISONIK-specific engineering process. This includes:

- Configuring parameters, e.g. for summer/winter compensation
- Creating time programs
- Text commentary
- Testing
- etc.

5.4 Status indication in the commissioning phase

NIDES indicators

This illustration shows a section of the NIDES.RX display:



The table below gives the meaning of the various states of the POWER (green) and STATUS (red) LEDs when commissioning the RX master:

Circumstances	LED	Description
Normal operation	POWER LED ON (continuous)	Power supply OK
Infolist download into NIDES (during normal operation)	Sequence: 1. POWER LED flashing 2. STATUS LED ON (continuous) 3. POWER LED stops flashing and reverts to continuous ON 4. STATUS LED OFF	– Infolist download in progress – Initialization in progress (RX database, values from RX master) – Infolist download complete – NIDES initialization complete
After a power failure	Sequence: 1. POWER and STATUS LEDs ON (continuous) 2. STATUS LED goes off POWER LED stays ON	– NIDES synchronizing database / fetching values from the controllers – Normal operation

Info on the BPS

No direct display with CFE versions prior to V2.1.

The start-up phase (Infolist download, generation of points, interrogation of all data points from the NIDES and interrogation of changes of value or state) ending when all RXC controllers and the associated data points in the BPS are reported as normal, generally takes a considerable time (in large projects up to ½ hour). During this period of time, the following options are available on the BPS to monitor progress:

- RUN 126, Diagnose 9 (communication status)
  indicates which controllers are already in the “Normal” state (i.e. RSta=0, FICT=1)
  Shows the communication status: "Normal" is indicated by KSta=0, NSta=1
  For more information on diagnostics, refer to the Engineering guide, document CA2Z8339en)
- COLBAS command ALR (Alarm Report) or ERP (Error Report):
  – Shows all points with an error status (display "?HW" or "not connected")
  If this display is blank, the start-up procedure is complete.

From CFE Version V2.1.1, the start-up procedure is displayed. Refer Technical principles the Engineering Guide for a description. This makes it unnecessary to use Diagnostics 9 or the ALR/ERP commands to check the process.

5.5 General options for operation

Operating the data points

The integrated data points can be operated with the VISOTOOL Editor, for example:

- To read the effective heating setpoint: @TCR1.ASPH
- To specify a new Comfort heating setpoint: @TCR1.SPCH:=20.5
  (basic setpoint setting for Comfort/Heating)

RMR report output

Room management reports can be obtained with the command RMR, for example:

RMR	for a report on all TEC rooms
RMR @ TCG n	for a specific TEC group
RMR @ TCR n	for a specific TEC room

Example of an RMR

The example below shows the command and the room management report for room TCR74. It will be seen from the report that the room has one RXC controller, TEC No. 39, for which the parameters and values are also shown:

```
rmr @tcr74
RMR        17:06:05                                               16-AUG-1999/MO

!FI TCR  $d070'TCR74  (T39)
              16-AUG-1999 17:04:18 AOPST=0
    SOPST=0                        APRST=0
    SOMOD=0                        ASTH =0                 VSPA =0
    MACT =0                        ASTC =0                 VEXA =0
    MOPST=2                        RRSC =0                 RWINO=0
+-----+------+-----+-----+-----+-----+-----+-----+------+------+------+------+
! TEC ! ARTP !OHWA !OCWA !OXWA !OHAIR!OCAIR!OHHB !OEAIR ! OFAN !AIRVS1!AIRVE !
!     !      !OHWA2!OCWA2!     !     !     !     !      !      !AIRVS2!      !
!     ! [C]  ! [%] ! [%] ! [%] ! [%] ! [%] ! [%] ! [%]  ![Step]![m3/h]![m3/h]!
+-----+------+-----+-----+-----+-----+-----+-----+------+------+------+------+
! 39  ! 27.2 !   0 ! 100 !   0 !   0 !   0 !   0 !    0 !    3 !    0 !    0 !
!     !      !   0 !   0 !     !     !     !     !      !      !    0 !      !
+-----+------+-----+-----+-----+-----+-----+-----+------+------+------+------+

RMR        17:06:08                                               16-AUG-1999/MO
```

Meaning of the parameters

A description of the various parameters can be found in the following documents:
CM2Z8303e "Process image BPS Point Types and Parameters" (FLN points) or
CM2Z8568en VISONIK Point Types and Parameter Description VVS20 (alphabetical)

Selective system report

The selective system report SYR /G is a valuable aid.
Enter "syr /g=74", for example, to list all points allocated to Room 74 @TCR74:

```
syr /g=74
SYR        17:06:21                                               16-AUG-1999/MO

!FI TCR  $d070'TCR74  (T39)
              16-AUG-1999 17:04:18 AOPST=0
!FI AO   $d070'AO39'29       iFanSpdSta
              01-JAN-1993 00:00:00 AO=3                      OPSTA=1

SYR        17:06:21                                               16-AUG-1999/MO
```

5.6 Structure of the TEC data points

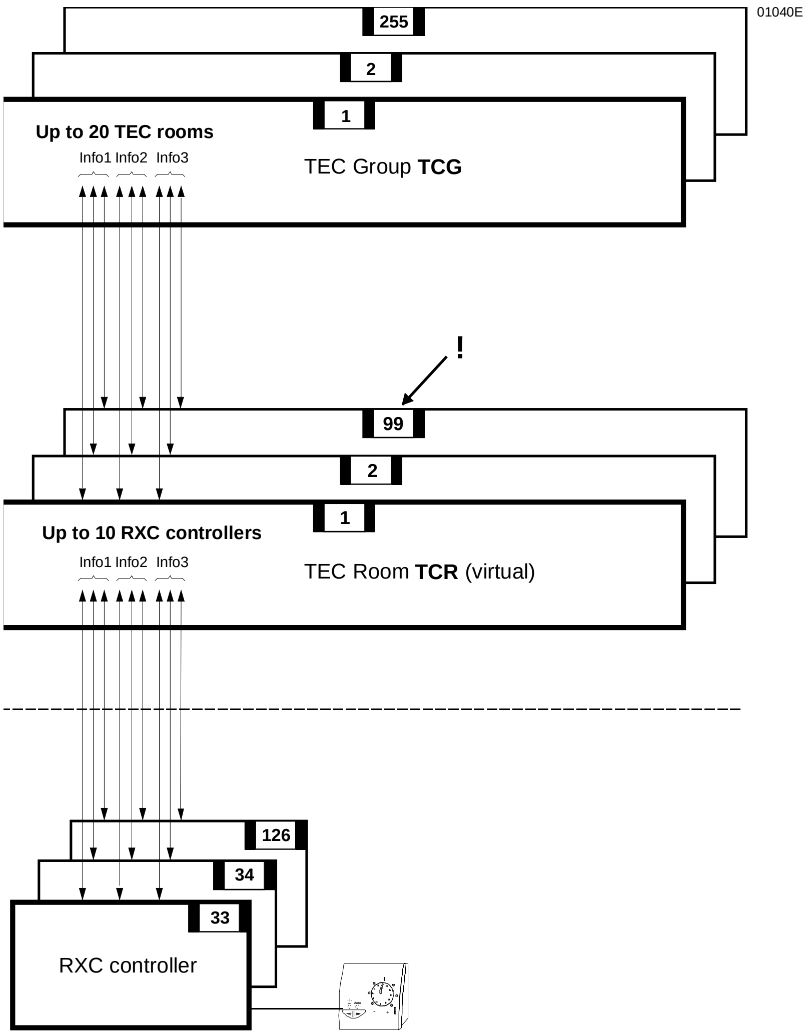
Introduction

For the purposes of integration into VISONIK, the RXC controllers are mapped to virtual TEC rooms as internal data points. Access to individual controllers takes place indirectly, via the room, as is the usual case for TEC controllers.

Addressing

The diagram below shows the structure for TEC addressing in VISONIK, with details of the address range and group size for:

– the TEC room (point type @TCR) and
– the TEC group (point type @TCG)



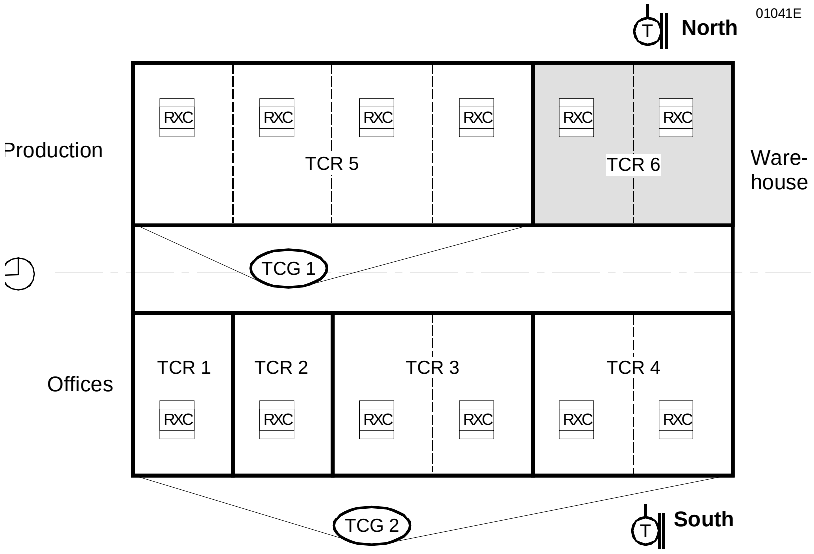
Note

For the integration of DESIGO RXC, the address range for the TEC rooms (TCR) is limited to 1 … 99 (instead of 1 … 255, as is usual for TEC).

5.7 Operation of the HVAC groups

Example

The diagram below again shows the example of the "Definition of HVAC groups"



Allocating the rooms to groups

The rooms are allocated to the groups in the GroupConfig.COL file. (Refer to the section on defining groups in the "Engineering" section). The CFE software converts the room allocations defined in the GroupConfig.COL to the room parameter TCR of the group points:

@TCGn.TCR(p)=m	n: HVAC group number (1..200)
	p: Index of HVAC room within HVAC group (1..20)
	m: Number of HVAC room (1..99)

The allocation of the rooms can be viewed with the COLBAS command
LIST @TCG / R.

Group functions

The following standard TEC functions are implemented throughout an entire TEC group (TEC master functions of the BPS)

– Statistical calculations (Min / Max / Mean room temperature and % share of operating modes)
– Setpoint correction based on outdoor temperature (summer/winter compensation, and see "Mapping functions", 3.1.11)
– OSC function (Optimum Start Control)
– The above are described in document CM2T8568en "System principles VVS20" under "TEC Points in the BPS – TEC Standard Functions".
–

Operating the groups

– The TEC group functions are operated via the following @TCG parameters:
– Summer/winter compensation parameters IWI / ISU / BSTP / SIDTO
– OSC parameters OSCA / OSTH / OSTC
– Operating mode defined for whole group SOPST / SOMOD
– Setpoint defined for whole group SPCH / SPCC etc.

Setpoint shift defined for whole group **@VIPn**

For more information on the parameters refer to document CM2Z8303e "BPS Process Image" under "FLN Points".

Note that the setpoint shift for the whole group is defined not via parameter RRSC, but via the @VIPn (main value VP) associated with the group @TCGn. The CFE software passes @VIPn to the RRSC of all rooms @TCRm belonging to the group.

5.8 Operating the blind and lighting groups

Introduction

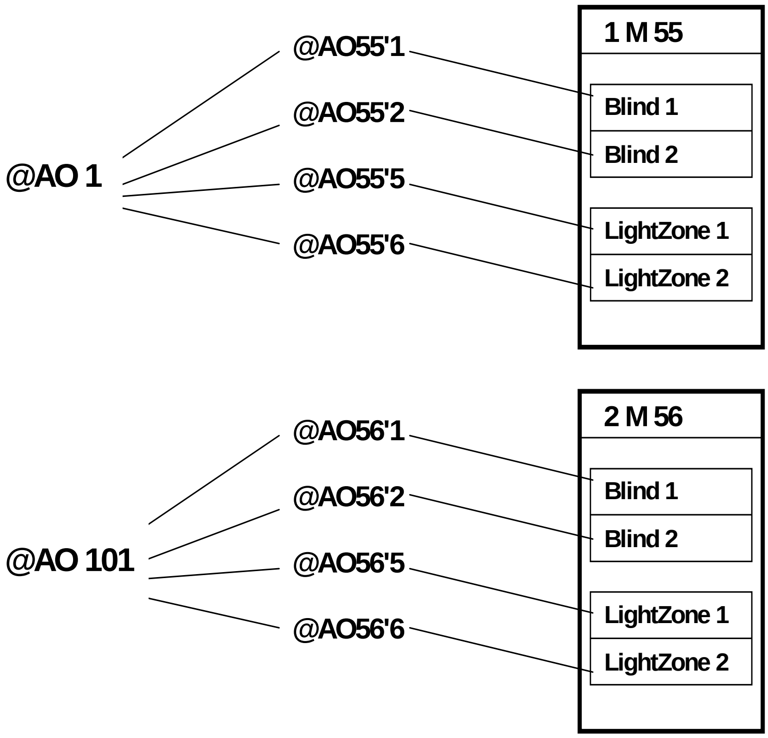
The RX master makes the group connections for blinds and lighting zones in accordance with the entries in the GroupConfig.col file (TSK 112).
For more information refer to "Group definition" in the "Engineering" section.

Example

The blinds of controllers 1M55 and 2M56 in Rooms 1 and 2 are to be assigned to a blind group, and the lighting zones of the same controllers to a lighting group (Priority level 3 for both groups). The entries in TSK 112 are as follows:

```
3000 DATA  1,3,55,56,-9
3001 DATA  –9
4000 DATA 101,3,55,56,-9
4001 DATA  -9
```

Based on these entries, the CFE software generates the group points **@AO1** (blind group) and **@AO101** (lighting group) and connects them to the data points for the blinds and lighting zones.



Operating the groups

The following parameters of the two group points @AOg (g=1 /101) can now be operated (R: **R**ead / W: **W**rite) :

Parameters	R / W	Description
@AO g.EMV (Specification of parameter EMV optional)	W	Command when g=1 (blinds): Up (0) / Down(1) / Down(2) / Auto(3) Command when g=101 (lighting zones): Off(0) Dimming (1…100%) Auto(255) [1]
@AO g.PARINT(1)	R	Last command issued [2]
@AO g.PARINT(2)	R / W	Operating mode: Local (0) / Remote(1)
@AO g.PARINT(3)	R	Priority (1..100)
@AO g.FBV	-	[3]

[1] The dimming command presupposes integration of the lighting commands (not possible with Light Override)

[2] EMV is immediately reset to –1 as soon as the command has been issued.

FBV changes together with EMV. There are no feedback signals for the group as a whole. The feedback signal must be inspected by the allocated link points. Read @AOx'y.FBV.

Transfer

The parameter values for the group commands and operating modes are passed on to the allocated link points @AOx'y for the blinds and lighting zones. These ensure the output to the relevant network variable. For details refer to Mapping functions 3.1.26 to 3.1.29.

Operating the blinds and lighting zones

These are operated directly via the parameters of the link point concerned.

Parameters	R / W	Description
@AOx'y.EMV	W	Command (see above)
@AOx'y.PARINT(1)	R	Last command issued [2]
@AOx'y.PARINT(2)	R / W	Mode (Local, Remote)
@AOx'y.PARINT(4)	Reserved	Internal memory. **Must not be used for other purposes**
@AOx'y.FBV	R	Feedback
@AOx'y.GROUP	R	Room number
@AOx'y.FICT	R	1 (Normal) / 0 (Alarm – not connected)

Note that the next group command will overwrite the last command issued to the individual blind or lighting zone.

6 Index

Siemens Switzerland Ltd.
Building Technologies Group
International Headquarters
Building Automation
Gubelstrasse 22
CH-6301 Zug
Tel. +41 41-724 24 24
Fax +41 41-724 35 22
www.sbt.siemens.com