Installer manual

# Integration of the By-me Plus system with KNX devices

By-me Plus SYSTEM

**KEY**

* The name of the dptx depends on the type of configuration; one of those possible is indicated.

** The dptx is not always configured; this depends on the functions to be implemented.

Dptx in the box : this is a system dptx, system clock.

# Index

**SECTION 1: GENERAL DESCRIPTION** from page 5

**SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL** from page 17

**SECTION 3: ENERGY MANAGEMENT** from page 57

**SECTION 4: TEMPERATURE CONTROL** from page 69

**SECTION 5: SOUND SYSTEM** from page 111

# Index

1 Introduction ..... 6
1.1 Purpose ..... 6
1.2 Audience and prerequisites ..... 6
1.3 Guide to reading this document ..... 6
2 Vimar BUS systems ..... 6
2.1 Communication objects ..... 6
2.2 Functional Blocks and Communication objects ..... 6
3 Installing a mixed By-me - KNX network ..... 6
3.1 How to calculate the group addresses from VIEW Pro ..... 7
3.2 How to calculate the scenario group addresses ..... 8
3.3 How to calculate the group addresses with ETS ..... 10
3.3.1 Calculating the group address for 3-level ETS visualization ..... 10
3.3.2 Calculating the group address for 2-level ETS visualization ..... 10
3.3.3 Setting the address in ETS ..... 10
4 Example of mixed network installation ..... 11
4.1 Example ..... 11
4.1.1 Determining the functions and communication objects ..... 11
4.1.2 Configuring the functions and groups with VIEW Pro ..... 12
4.1.3 Measuring the values of the group addresses with VIEW Pro ..... 12
4.1.4 Configuring the EIB and KNX devices using ETS ..... 12
5 Instructions for SCADA programming ..... 13
5.1 Instructions for setting up SCADA: what not to do. ..... 13
5.2 SCADA settings: how to work. ..... 14

# SECTION 1: GENERAL DESCRIPTION

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to clarify how VIMAR By-me devices can be used in KNX networks. By-me devices are described in terms of their KNX network interface, indicating how to connect them to the KNX bus and how to ensure interoperability with standard KNX S-Mode devices supplied by different manufacturers, and explaining how to interface the VIMAR devices by configuring the devices of other manufacturers via the ETS software tool.

### 1.2 Audience and prerequisites

The audience for this document are all engineers and designers intending to integrate the VIMAR By-me system with KNX devices. The prerequisite is being familiar with the fundamentals of the KNX communication standard and the ETS software installation programme.

### 1.3 Guide to reading this document

The document aims to be a reference tool and practical guide for anyone planning to build a mixed device network, composed of KNX and VIMAR By-me devices.

The first part, consisting of chapter 2 only, offers a short introduction to the general concepts of the KNX bus and the specifics of VIMAR By-me, helping to understand the rest of the document.

The second part, chapters 3 and 4, is more practical, describing which operations have to be done to integrate By-me devices with KNX devices, and gives a complete practical example of a mixed installation using ETS.

## 2. Vimar BUS systems

The VIMAR By-me system includes a complete series of devices covering all the various needs of a home automation system. In particular, it is possible to automatically control lights and roller shutters, loads and manage energy, temperature control and sound systems.

All devices are able to ensure the CO-EXISTENCE with KNX devices, i.e. with no interference between one device and another (ref. KNX - Volume 3: system specifications, Part 1: architecture, Chapter 2: Glossary) and cooperation. This goes for both the electrical level and the level of information exchange on the KNX network.

Please consult the VIMAR manuals for a description of each device in terms of its functions, the way it is connected to the BUS cable and the current consumption from the BUS, to be able to calculate and correctly size the power supply system.

### 2.1 Communication Objects

VIMAR By-me devices are able to communicate with standard KNX devices via standard “Datapoints” defined by the KNX standard “Interworking” model. The DATAPOINTS are implemented as GROUP OBJECTS (GO), as defined by the KNX architecture.

The DATAPOINTS in KNX systems are described in the ETS design and installation programme as objects of the devices. In the rest of the document, the DATAPOINTS are indicated as OBJECTS, according to the terminology used by ETS.

As regards the definitions and details of the objects, refer to the document Volume 3: System specifications, Part 7: Interworking, Chapter 2: Datapoint types.

### 2.2 Functional Blocks and Communication objects

A FUNCTIONAL BLOCK consists of one or more functions which cannot be separated and which are together in the same device. These functions are created using input and/or output communication OBJECTS.

Each By-me device is composed of one or more FUNCTIONAL BLOCKS.

For example, the Vimar device 20526 (rocker button controls and actuator) is made of 3 FUNCTIONAL BLOCKS:

- FB_V_ROCKER_BUTTONS (LH button)
- FB_V_ROCKER_BUTTONS (RH button)
- FB_V_ACTUATOR (Relay Actuator)

The following chapters describe in detail all Vimar By-me devices, indicating their functional blocks and the communication objects present in each functional block.

## 3. Installing a mixed By-me - KNX network

Use the VIMAR devices to make them cooperate with devices conforming to the KNX standard is possible, we just need to know which OBJECTS the required functions are conveyed on and then link them to the OBJECTS in the KNX devices.

To link VIMAR OBJECTS to the OBJECTS of KNX devices, the following are required:

- The objects must be of the same type, i.e. they have the same DATPOINT_TYPE
- They must use the same GROUP ADDRESS

First of all it is therefore necessary to identify the By-me OBJECTS to be used, identify the By-me group addresses and use the same group addresses in the OBJECTS on the KNX device.

## SECTION 1: GENERAL DESCRIPTION

The following paragraphs explain in detail the operations necessary to create this link.

The range of KNX addresses used by the control unit to configure the By-me devices varies according to the “Group Offset” parameter which is set in the VIEW Pro app during the first configuration flow.

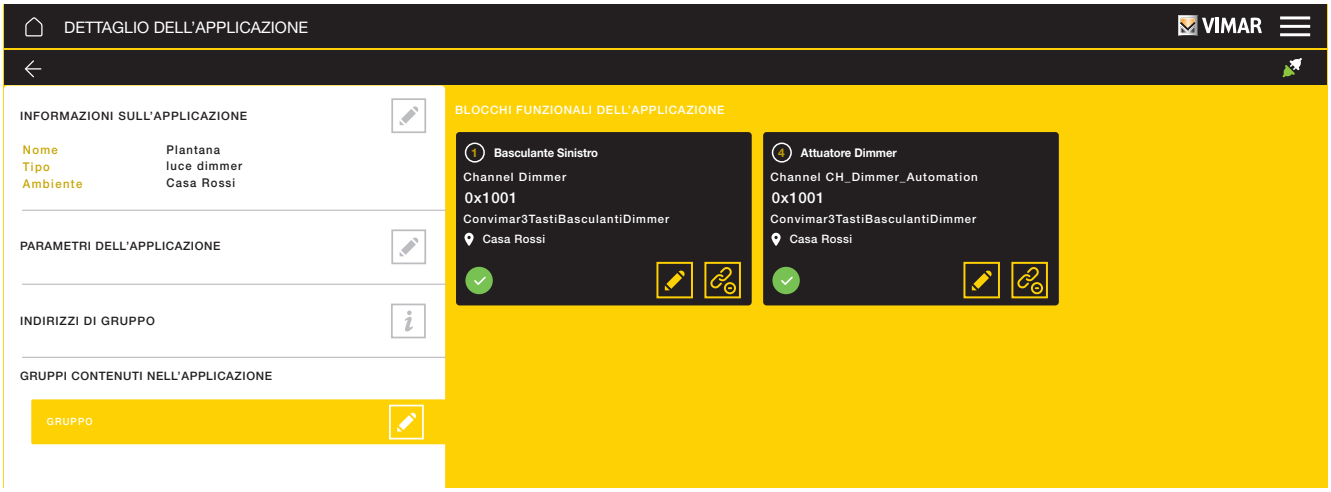
| Group Offset | Address range [KNX 3-level format] | Address range [hexadecimal format] |
|---|---|---|
| 0x0B00 | from 1/3/0 to 3/7/255 | from 0x0B00 to 0x1FFF |
| 0x2000 | from 4/0/0 to 7/7/255 | from 0x2000 to 0x3FFF |
| 0x4000 | from 8/0/0 to 11/7/255 | from 0x4000 to 0x5FFF |
| 0x6000 | from 12/0/0 to 15/7/255 | from 0x6000 to 0x7FFF |
| 0x8000 | from 16/0/0 to 19/7/255 | from 0x8000 to 0x9FFF |
| 0xA000 | from 20/0/0 to 23/7/255 | from 0xA000 to 0xBFFF |
| 0xC000 | from 24/0/0 to 27/7/255 | from 0xC000 to 0xDFFF |
| 0xE000 | from 28/0/0 to 31/7/255 | from 0xE000 to 0xFFFF |

**Important: whatever the group offset used, the address range from 0/0/0 to 1/3/0 should always be kept free.**

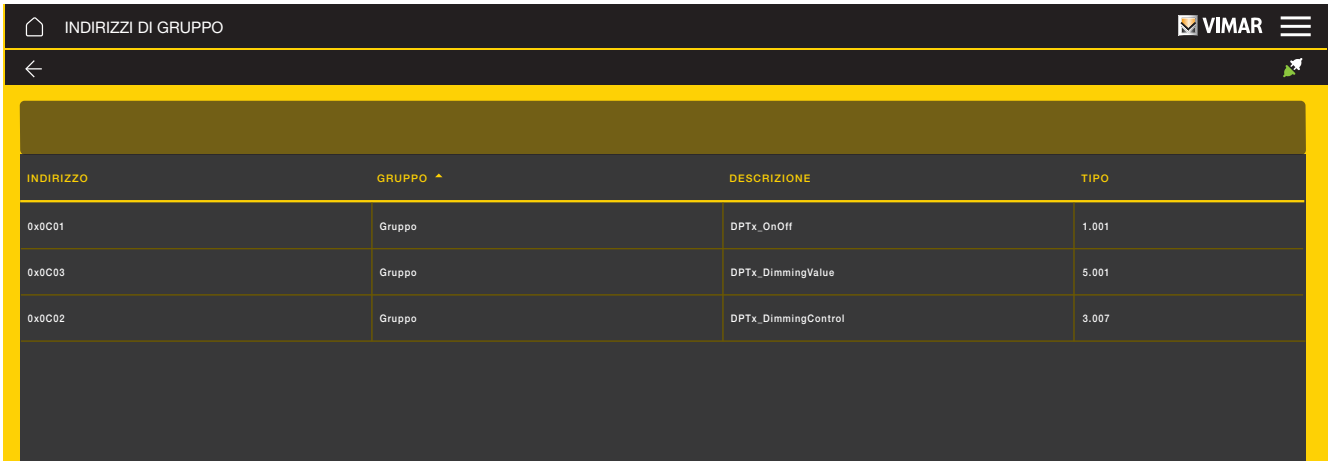
### 3.1 How to calculate the group addresses from VIEW Pro

To display and therefore use the group addresses in VIEW Pro, proceed as follows:

- In the APPLICATIONS screen, display the applications created.
- Click on next to the one required to display the detailed screen of the application.




- Click on to display the list of group addresses.

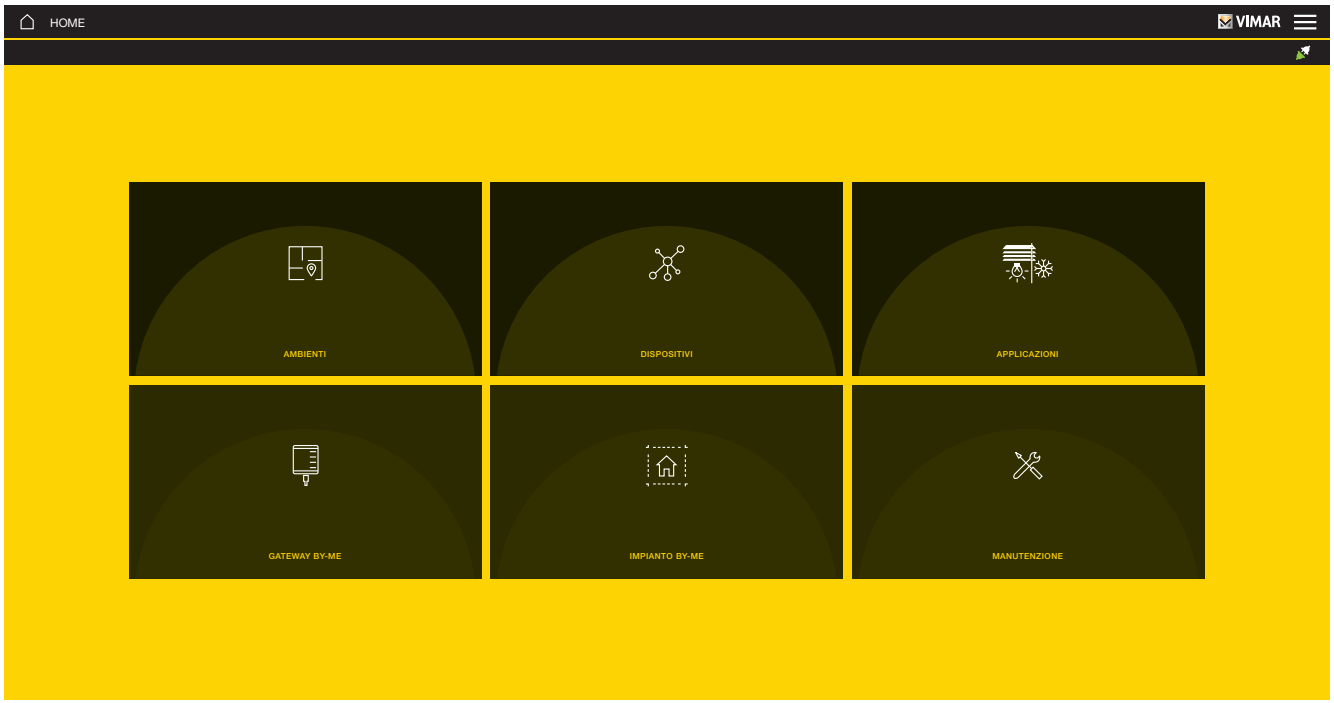


## SECTION 1: GENERAL DESCRIPTION

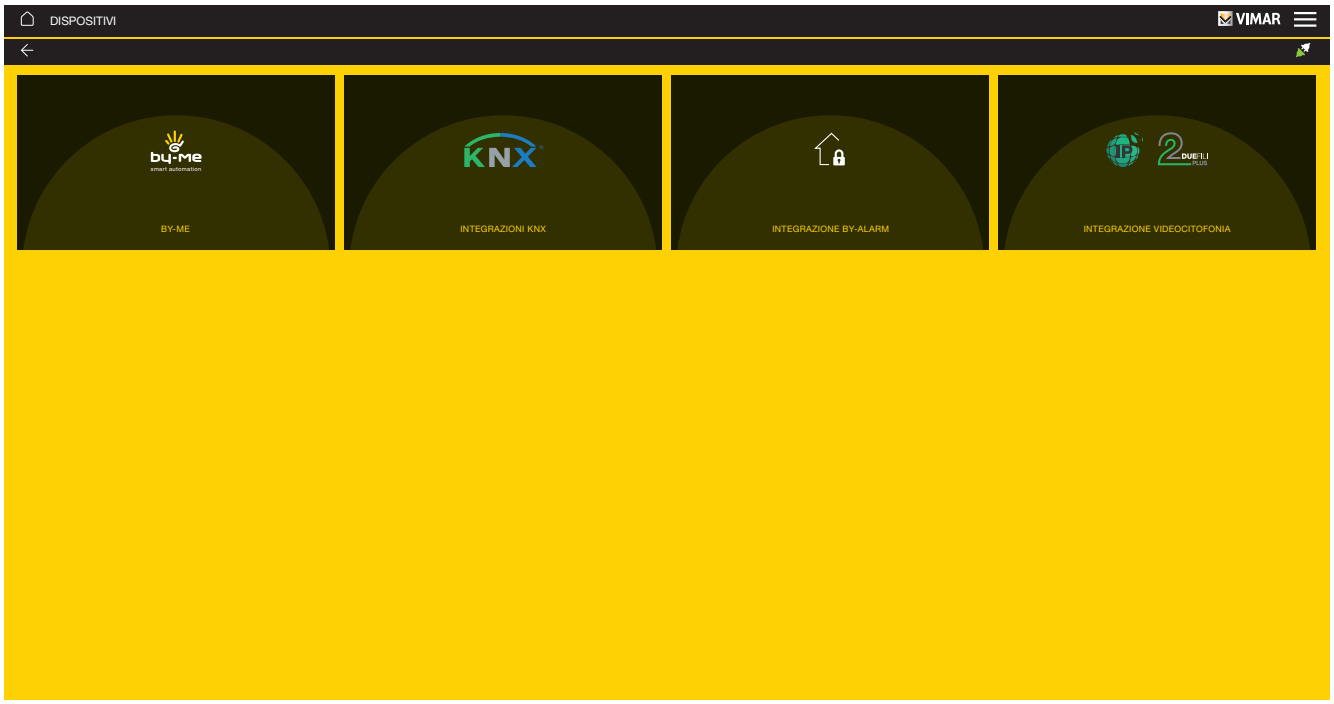
### 3.2 How to calculate the group addresses of the scenarios

In VIEW Pro, the group addresses of the scenarios are calculated as follows:

- Click on DEVICES:



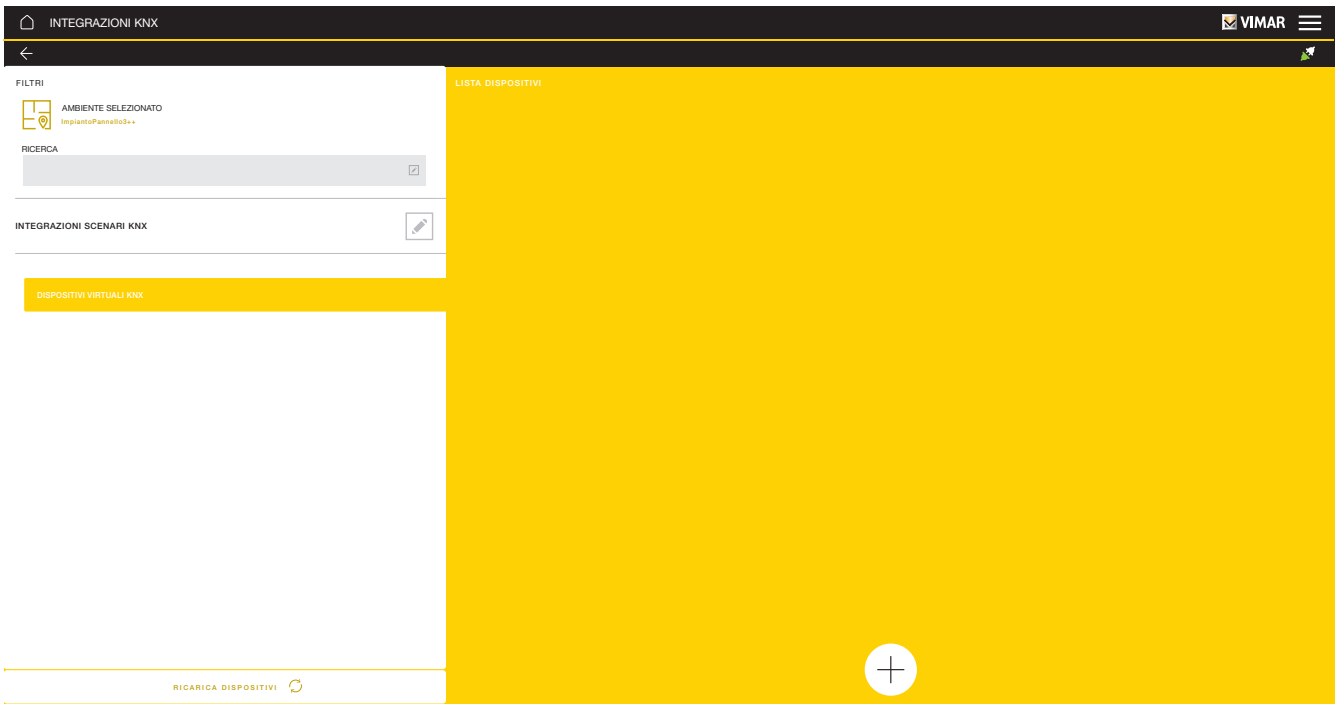

- Click on KNX INTEGRATIONS:



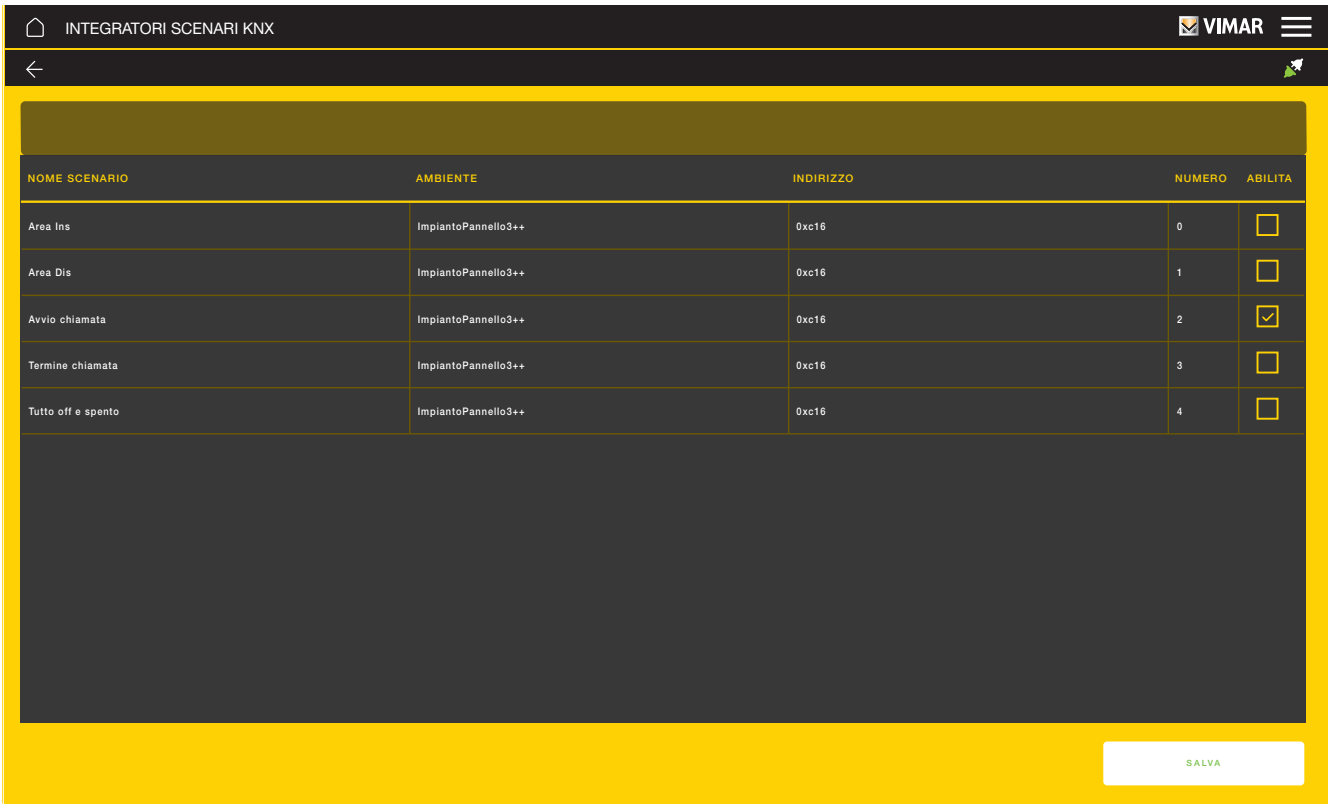
## SECTION 1: GENERAL DESCRIPTION



- Click on KNX SCENARIO INTEGRATIONS:




The list of configured scenarios is displayed; each scenario is paired to a group address and a scenario number. If you wish to use a scenario for and integration with a KNX device you must select ☑ next to the one required.




**IMPORTANT:** For By-me integration with ETS it is not possible to use to use ETS scenario 1. So if you create a scenario in By-me, to scenario to pair for ETS devices will have an index equal to that of By-me + 1.

Example: By-me scenario 5 will correspond to ETS scenario 6.

# SECTION 1: GENERAL DESCRIPTION

### 3.3 How to set the group addresses with ETS

Once the By-me group address has been detected, it must be assigned to an object on a KNX device using the ETS programme. ETS has two ways of visualising addresses: you can choose between 2 or 3 levels and this changes the display of the addresses during the design phase, and changes the way of calculating the value of the By-me group address to be entered.

ETS3 automatically adopts the three-level visualization.

The group addresses are displayed with three decimal places separated by points when selecting “Group address levels = 3 levels”. For example the address 0B0A must be entered as 1.3.10.

When visualizing with two levels, on the other hand, the group addresses are displayed with two decimal places separated by points when selecting “Group address levels = 2 levels”. For example the address 0B0A must be entered as 1.778.

#### 3.3.1 Calculating the group address for 3-level ETS visualization

To calculate the value to enter as a group address we need to divide the hex value into two octets, which represent the most and least significant parts of the value. For example, 0B0A is composed of a most significant part, 0D, and a least significant part, 0A.

The least significant part is translated directly into a decimal to obtain the last component of the address, the third level.

In our case: least significant part = 0A, decimal value = 10. The last component of the address is 10.

To transform a number from the hex notation to the decimal notation we can use the Windows calculator: set the scientific mode, select “Hex” and enter the amount. Then select “Dec” and you will obtain the same number in decimal format. E.g.: 0B0A corresponds to 2826.

The most significant part of the address must again be divided into two parts: we need to transform it into binary notation. Again using the Windows calculator in scientific mode, set the notation to (HEX), enter the number and select BIN. The number is represented by a sequence of ones and zeros. For example 0B becomes 1011.

To obtain the central part of the address, the second level, we must take only the last 3 numbers of the most significant part expressed in binary notation and transform it into decimal notation.

In our case: most significant part = 0B, binary value = 1011. We consider only the last three numbers = 011, transformed in decimal = 3. The second level of the address is 3.

To obtain the first level of the address, we must take the remaining numbers having removed the last 3 numbers of the most significant part in binary notation and transform them into decimal notation.

In our case: most significant part = 0B, binary value = 1011. We eliminate the last three numbers = 1, transformed into decimal = 1. The first level of the address is 1.

The address 0B0A represented on three levels is 1/3/10.

#### 3.3.2 Calculating the group address for 2-level ETS visualization

To calculate the value to enter as a group address we need to divide the hex value into two octets, which represent the most and least significant parts of the value. For example, 0B0A is composed of a most significant part, 0D, and a least significant part, 0A.

The most significant part of the address must again be divided into two parts: we need to transform it into binary notation. Using for example the Windows calculator in scientific mode, set the notation to (HEX), enter the number and select BIN. The number is represented by a sequence of ones and zeros. For example 0B becomes 1011.

To obtain the first level of the address, eliminate the last 3 numbers and take the remaining numbers expressed in binary notation and transform them into decimal notation.

In our case: most significant part = 0B, binary value = 1011. We eliminate the last three numbers = 1, transformed into decimal = 1. The first level of the address is 1.

To obtain the central part of the address, we must take only the last 3 numbers of the most significant part expressed in binary notation and transform it into hex notation. The number obtained is the most significant part of the second level. We join the most significant part with the least significant part obtained beforehand and transform it all into a number expressed in decimal notation.

In our case: most significant part = 0B, binary value = 1011. We consider only the last three numbers = 011, transformed into hex = 3. The least significant component is 0A, together they are 030A which expressed in decimal notation is 778.

The address 0B0A represented on two levels is 1/778.

#### 3.3.3 Setting the address in ETS

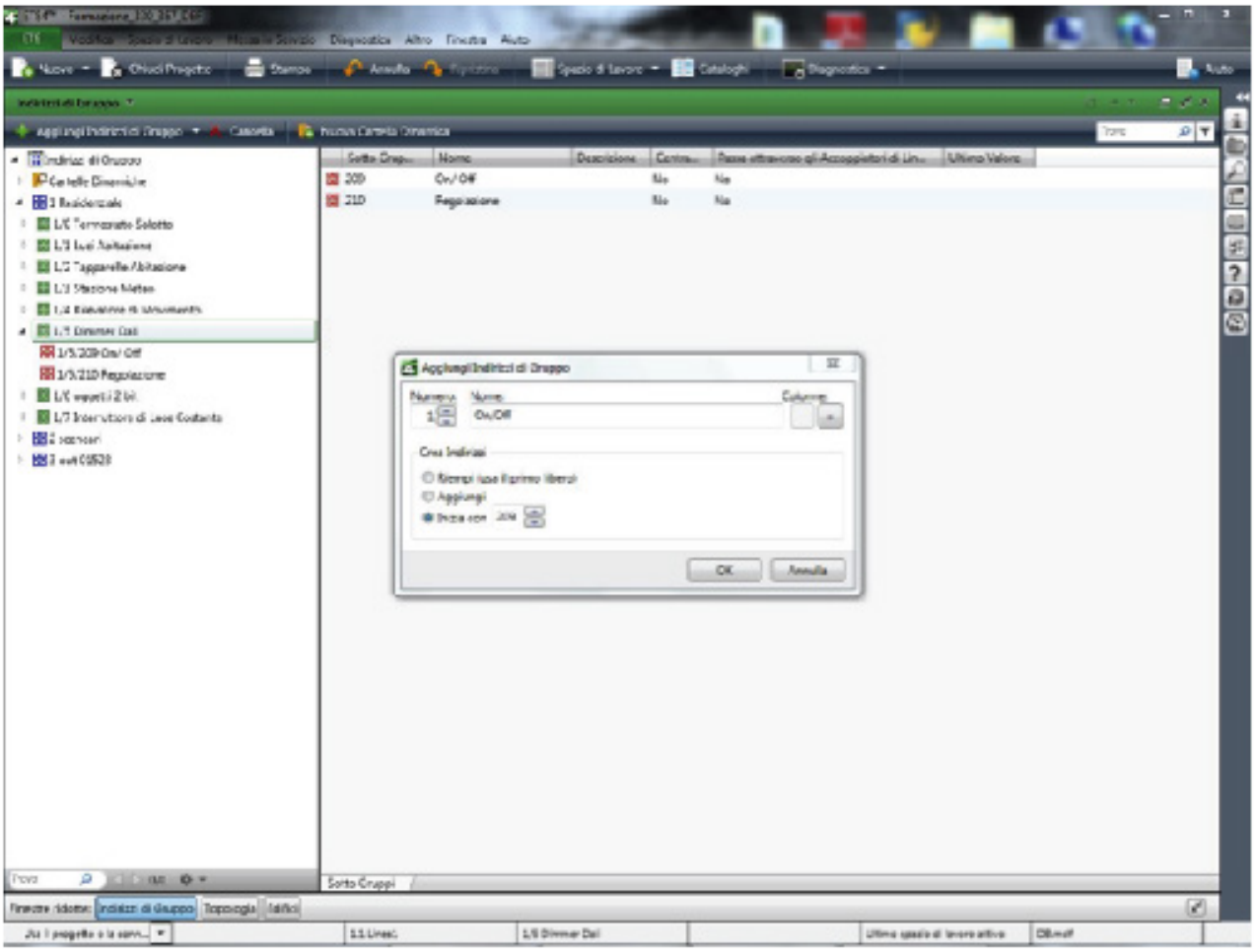
Having obtained the address in the two- or three-level visualization mode, it is then possible to assign it to the OBJECT of the KNX device you want to integrate into the By-me system.

From the main branch of Group Addresses, select "Add Main Group Address"

Select the address of the Main Group you have just created and select "Add Intermediate Group Address"

Select the address of the Intermediate Group you have just created and select "Add Group Address" to complete the creation of the desired group; the image below is an example of creating the group 1/5/209.

## SECTION 1: GENERAL DESCRIPTION



## 4. Example of mixed network installation

To facilitate understanding, here is illustrated an example of installation with interaction between some By-me devices and some VIMAR KNX devices.

It is important to carry out the installation phases in the correct order.

1) Determining the functions and communication objects.
2) Configuring the function groups with VIEW Pro.
3) Measuring the values of the group addresses created in VIEW Pro to use for the groups which include devices which do not belong to the By-me system.
4) Configuring the EIB and KNX devices using the ETS installation programme.

VIEW Pro assigns the group addresses to the various DATAPOINTS which are configured in the system, it is therefore indispensable that the By-me system be installed first.

**It is not possible to adapt the group addresses of a By-me installation to other group addresses that are predetermined with ETS.**

### 4.1 Example

The following example uses the Vimar DALI gateway article 01544 to switch on/off and adjust the brightness of a DALI dimmer with a By-me rocker button article 20541.

#### 4.1.1 Determining the functions and communication objects

To control a DALI dimmer with the Vimar DALI gateway article 01544 to switch on/off and change the brightness (Dimming Control) you need to use the following communication objects, available on ETS:

0: Channel A - On/Off (data type Switch 1.001)

1: Channel A - Relative Dimmer (data type Dimming Control 3.007)

As indicated in the descriptive model of the By-me rocker button (see chapter 3.3 of the document "*By-me VIMAR – KNX: Guide for integrating light and roller shutter control*"), if set as a Dimmer the By-me button uses the same communication objects as the DALI gateway described above, and is therefore compatible with this use.

## SECTION 1: GENERAL DESCRIPTION

#### 4.1.2 Configuring the functions and groups with VIEW Pro

Through the procedures illustrated in pars. 4.2 and 4.3 of the By-me Plus system manual, create the applications and configure the relative functions.

#### 4.1.3 Measuring the values of the group addresses with VIEW Pro

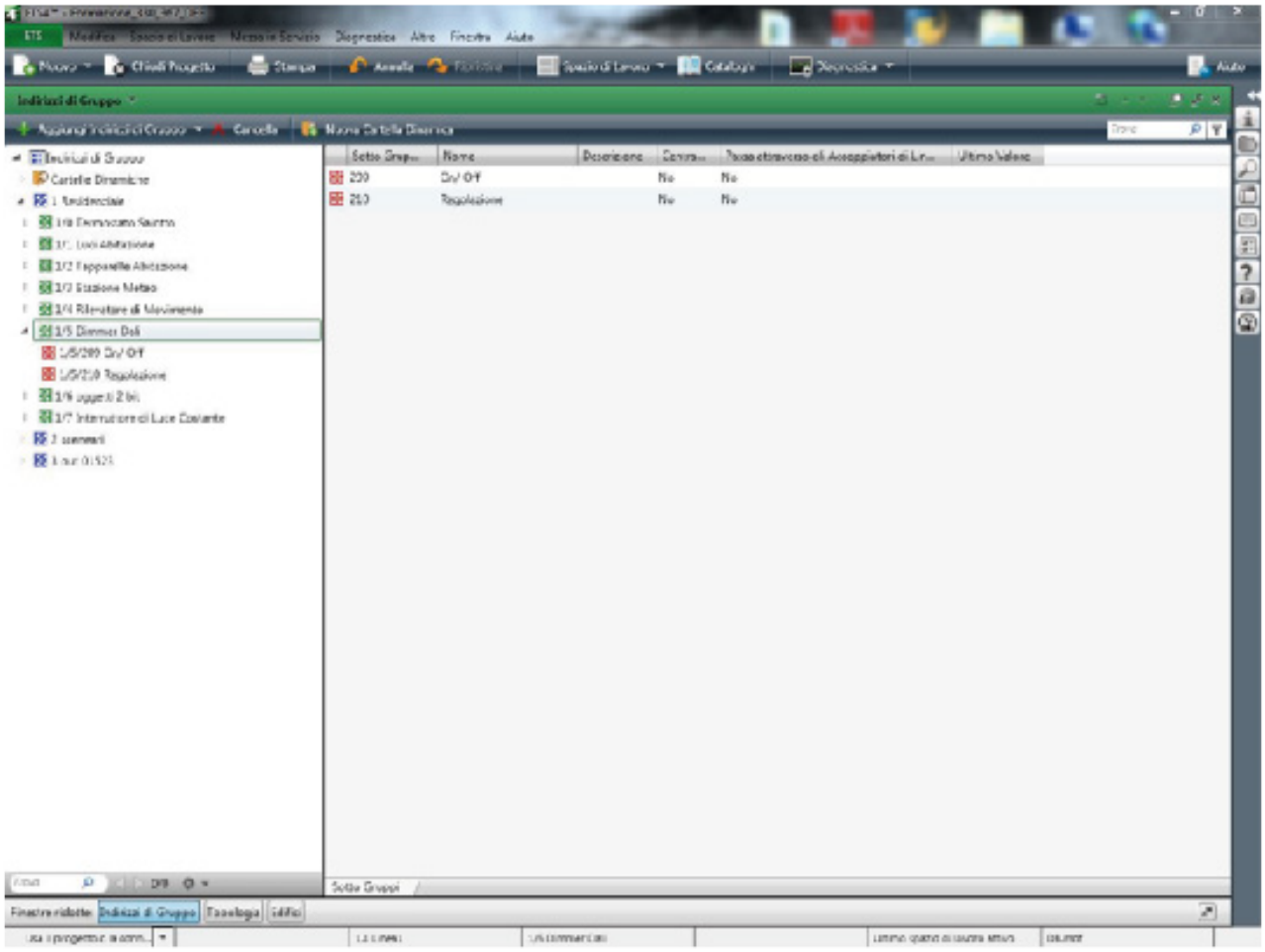
For all involved applications, display the value of the group addresses which must then be used in the KNX integration.

#### 4.1.4 Configuring the EIB and KNX devices using ETS

The KNX and EIB devices must be configured using the ETS programme. We must create or open an existing project and enter the devices, selecting the application suited to the required function.

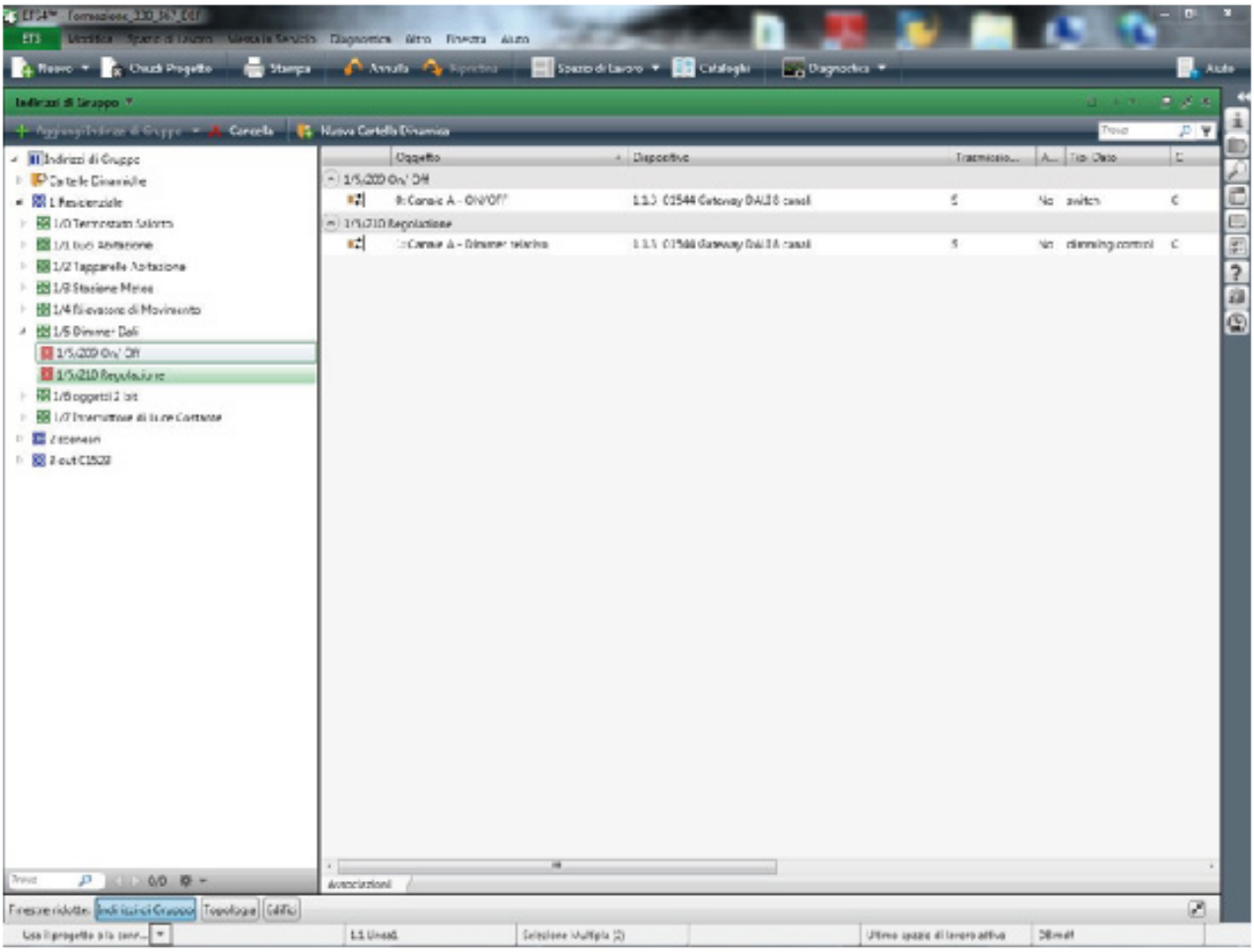
To control a DALI dimmer with a By-me button you need to configure two groups, one for the on/off function (OnOff) with address 1/5/209, and the other for the adjustment (Dimming Ctrl) with address 1/5/210.

The following figure shows the ETS 4 screen with the example project and the two required groups.



The following figure shows the pairing of the DALI gateway communication objects with the newly created groups; finish the configuration on ETS and download it to the KNX device.

## SECTION 1: GENERAL DESCRIPTION



The integration is thus concluded, both the By-me button and the KNX Gateway use the same group addresses, and so they can communicate over the bus to be able to interact with each other correctly.

## 5. Instructions for SCADA programming

It is possible to use commercial SCADA (Supervisory Control and Data Acquisition) software programmes with the KNX BUS interface to monitor VIMAR devices.

These software programmes use the KNX group addresses in reading and writing to view or edit the status of the KNX network they control.

The indications given in the previous chapter describe the system to understand all the group addresses used for the operation of the By-me network, once the system has been installed and programmed and is therefore ready to be controlled using SCADA. The following paragraphs provide instructions for the correct interaction of the SCADA software with a generic KNX network.

### 5.1 Instructions for setting up SCADA: what not to do.

In contrast to other automation systems, the KNX BUS is an EVENTS BUS. This means that the protocol structure is optimised to carry the status change information of the various sensors, detectors, buttons etc. to the actuators and information users.

In this way the information is guaranteed to reach its recipient in the fastest time possible.

The KNX BUS is not a BUS based on the periodical request or POLLING system to manage information, and indeed this type of activity tends to place the KNX communication system in crisis. A SCADA which relies heavily on the polling system would cripple a KNX automation system, reducing performance or even impeding operations.

For example, in an office it could be difficult to switch on the lights, or in a hotel system it could be impossible to open the door to the room as the client's transponder card cannot be read,as the opening message cannot "get through" due to the heavy traffic on the BUS.

The consequences are easy to imagine: the user gets the feeling that the system does not work. The first to be deemed responsible for this type of situation are the device suppliers, i.e. VIMAR who supplies the By-me system.

To conclude: DO NOT use POLLING to display the system status.

## SECTION 1: GENERAL DESCRIPTION

### 5.2 SCADA settings: how to work.

If we wish to provide a synoptic picture which displays the status of the different devices we must use the “passive” reading method and status conservation. The programme must wait for all the telegrams which are sent on the BUS and must save the received status. When the programme has to display the status of some devices, it recovers the information from its internal memory without requesting a reading on the BUS.

There is a problem of initialising the programme status memory, and this must be resolved by reading all the statuses with a known, "slow" frequency without congesting the BUS with status request messages and therefore replies from the devices.

It is clear that there are systems in which the server on which the scada software is installed is never switched off, therefore the device status is always present, and the initialisation problem is not important, when it is switched on we can wait for all statuses to be read passively one by one.

There are also "small" systems in which it is simpler to make all the devices carry out a transition which allows the SCADA to know the status of the system without the need for initialisation.

In the vast majority of cases, however, we have to include an initialisation procedure which recovers all the status information on the various devices from the BUS, however diluting the requests over time.

Finally, in some cases it may be necessary to read the status of some devices periodically; in this case it is important to include few devices with this functionality on the same visualization page, reading their status on the BUS with low frequency, in terms of minutes.

| Table of rules for SCADA usage with KNX BUS systems | |
|---|---|
| Function to be performed | Method |
| Status synoptic visualization | BUS listening and saving of the status of all groups. When a page is called up that displays the device status, the status is read by the programme memory and not by the KNX BUS. |
| Recovery of missing statuses | This operation must be done frequently (i.e. every minute) checking that the devices displayed on the same page do not send their request at the same time. |
| Polling needs | If it is STRICTLY necessary to make periodical requests, these must be FEW and frequent (i.e. every minute) |

Using these basic rules you are sure to obtain both an accurate display of the status of the system being controlled and a guarantee that the normal system operation will not be disturbed.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

# Index

1 Introduction ..... 20
1.1 Purpose ..... 20
2 Composition of By-me devices ..... 20
2.1 Single push button controls, 16960 and 14520 and 20520 ..... 20
2.2 Single push button controls and actuator, 16965 and 14525 and 20525 ..... 20
2.3 Rocker button controls, 16961 and 14521 and 20521 ..... 20
2.4 Rocker button controls and actuator, 16966 and 14526 and 20526 ..... 20
2.5 Rocker button controls and roller shutter actuator, 16967 and 14527 and 20527 ..... 20
2.6 Rocker button controls and actuator for slave controls, 16968 and 14528 and 20528 ..... 20
2.7 3-push button controls, 16980 and 14540 and 20540 ..... 20
2.8 3-push button controls and actuator, 16985 and 14545 and 20545 ..... 20
2.9 3-rocker button controls, 16981 and 14541 and 20541 ..... 20
2.10 3-rocker button controls and actuator, 16986 and 14546 and 20546 ..... 20
2.11 3-rocker button controls and roller shutter actuator, 16987 and 14547 and 20547 ..... 20
2.12 3-rocker button controls and actuator for slave controls, 16988 and 14548 and 20548 ..... 21
2.13 Presence detector, 16935 and 14485 and 20485 ..... 21
2.14 Interfaces for traditional controls, 16955 and 14515 and 20515 ..... 21
2.15 Interfaces for traditional controls 1M, 16958 and 14518 and 20518 ..... 21
2.16 Actuators with relay output, 16975 and 14535 and 20535 ..... 21
2.17 Actuators with relay output 1M, 16974 and 14534 and 20534 ..... 21
2.18 DIN – Actuator with relay output, 1 relay, 01850.1 ..... 21
2.19 DIN – Actuator with relay output, 4 relay, 01851.1 ..... 21
2.20 DIN – Actuator with relay output for two roller shutters, 01852 ..... 21
2.21 DIN – Actuator with relay output for two roller shutters, 01852.2 ..... 21
2.22 DIN – Actuator with relay output for ballast, 01856 ..... 21
2.23 DIN – DIN – Master dimmer, 01853 ..... 21
2.24 DIN – MASTER dimmer 230V, 01870 ..... 21
2.25 DIN – By-me LED RGB power supply unit, 01877 ..... 21
2.26 DIN – Two-rocker button controls and SLAVE actuator, 16529,14529 and 20529 ..... 21
2.27 DIN – Three-rocker button controls and MASTER actuator, 16529, 14529 and 20529 ..... 22
2.28 1-10Vdc actuator LED 120-230V MARINE, 01975 ..... 22
2.29 1-10Vdc actuator LED 12-24V MARINE, 01976 ..... 22
2.30 PWM actuator for 12-24V LED relay MARINE, 01978 ..... 22
2.31 Four-button home automation control, 21520 ..... 22
2.32 Four-button home automation control, 21520.1 ..... 22
2.33 6-button home automation control, 21540 ..... 22
2.34 6-button home automation control, 21540.1 ..... 22
2.35 Multi-function home automation module 9in 8out relay, 01470 ..... 22
2.36 3in 3out LED home automation module, 01475 ..... 22
2.37 2in 3out home automation module (1 roll-shut+2 LEDs), 01476 ..... 22
2.38 2in 3out home automation module(1 relay+2 LEDs), 01477 ..... 23
2.39 DIN – Multifunction home automation actuator 4out relay, 01471 ..... 23
2.40 2M 4-button home automation control, 01480 ..... 23
2.41 2M 4-button home automation control + actuator, 01481 ..... 23
2.42 2M 4-button home automation control + roller shutter+slats, 01482 ..... 23
2.43 3M 6-button home automation control, 01485 ..... 23
2.44 3M 6-button home automation control + actuator, 01486 ..... 23
2.45 3M 6-button home automation control + roller shutter+slats, 01487 ..... 23
2.46 Home automation actuator+RGBW 4OUT controller, 01417 ..... 23
2.47 Home automation controller 2OUT200W LED120-240V, 01418 ..... 24
2.48 4-button home automation control+ LED controller 240V 2M, 01488 ..... 24
2.49 4-button home automation control+controller 0/1-10V 2M, 01489 ..... 24
3 Description of By-me device functional blocks ..... 24
3.1 Functional block “VIMAR Rocker Buttons” ..... 24
3.2 Functional block “VIMAR Rocker Buttons”: On Off control configuration ..... 25
3.2.1 Detailed description of communication Datapoints ..... 25
3.2.1.1 OnOff (output) ..... 25
3.2.1.2 OnOff (input) ..... 25
3.3 Functional block “VIMAR Rocker Buttons”: Dimmer Control Configuration ..... 26
3.3.1 Detailed description of communication Datapoints ..... 26
3.3.1.1 OnOff (output) ..... 26
3.3.1.2 OnOff (input) ..... 26
3.3.1.3 Dimming Ctrl ..... 26
3.4 Functional block “VIMAR Rocker Buttons”: Roller shutter control configuration ..... 26
3.4.1 Detailed description of communication Datapoints ..... 26
3.4.1.1 StopStep_UpDown ..... 26
3.4.1.2 Move_UpDown ..... 27

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL


3.5 Functional block “VIMAR Rocker Buttons”: Scenario control configuration ..... 27
3.5.1 Detailed description of communication Datapoints ..... 27
3.5.1.1 Scene 1 ..... 27
3.5.1.2 Scene 2 ..... 27
3.6 Functional block “VIMAR Single push buttons” ..... 28
3.7 Functional block “VIMAR Single push buttons”: On Off control configuration ..... 28
3.7.1 Detailed description of communication Datapoints ..... 28
3.7.1.1 OnOff (output) ..... 28
3.7.1.2 OnOff (input) ..... 28
3.8 Functional block “VIMAR Single push buttons”: Scenario control configuration ..... 28
3.8.1 Detailed description of communication Datapoints ..... 29
3.8.1.1 Scene 1 ..... 29
3.9 Functional block “VIMAR contact interface” ..... 29
3.9.1 Detailed description of communication Datapoints ..... 30
3.9.1.1 OnOff ..... 30
3.10 Functional block “VIMAR actuator” ..... 30
3.10.1 Detailed description of communication Datapoints ..... 30
3.10.1.1 OnOff ..... 30
3.11 Functional block “VIMAR Roller shutter actuator” ..... 31
3.11.1 Detailed description of communication Datapoints ..... 31
3.11.1.1 StepStop_UpDown ..... 31
3.11.1.2 Move_UpDown ..... 31
3.12 Functional block “VIMAR Roller shutter-Slats actuator” ..... 31
3.12.1 Detailed description of communication Datapoints ..... 32
3.12.1.1 StopStep_UpDown ..... 32
3.12.1.2 Move_UpDown ..... 32
3.13 Functional block “VIMAR DIMMER actuator” ..... 32
3.13.1 Detailed description of communication Datapoints ..... 33
3.13.1.1 OnOff ..... 33
3.13.1.2 Dimming Value ..... 33
3.13.1.3 Dimming Ctrl ..... 33
3.14 Functional block “VIMAR Infrared presence detector” ..... 33
3.14.1 Detailed description of communication Datapoints ..... 33
3.14.1.1 OnOff ..... 33
3.15 Functional block “VIMAR White Dimmer Actuator” ..... 34
3.15.1 Detailed description of communication Datapoints ..... 34
3.15.1.1 OnOff ..... 34
3.15.1.2 Relative Set Value ..... 34
3.15.1.3 Absolute Set Value ..... 34
3.15.1.4 Info OnOff ..... 34
3.15.1.5 Info Value ..... 34
3.16 Functional block “VIMAR RGB Dimmer Actuator” ..... 35
3.16.1 Detailed description of communication Datapoints ..... 35
3.16.1.1 OnOff ..... 35
3.16.1.2 Relative Set Value ..... 35
3.16.1.3 Absolute Set RGB Value ..... 35
3.16.1.4 Relative Set Hue ..... 35
3.16.1.5 Relative Set Saturation ..... 36
3.16.1.6 Switch FadingShow ..... 36
3.16.1.7 Info OnOff ..... 36
3.16.1.8 Info FadingShow ..... 36
3.16.1.9 Info Absolute RGB Value ..... 36
3.17 Functional block “VIMAR RGB and White Dimmer Actuator” ..... 36
3.18 Functional block “VIMAR RGB and White Dimmer Actuator”: White Control Configuration ..... 37
3.18.1 Detailed description of communication Datapoints ..... 37
3.18.1.1 OnOff ..... 37
3.18.1.2 Relative Set Value ..... 37
3.18.1.3 Absolute Set Value ..... 37
3.18.1.4 Info OnOff ..... 37
3.18.1.5 Info Absolute Value ..... 37
3.19 Functional block “VIMAR RGB and White Dimmer Actuator”: Colour Control Configuration ..... 37
3.19.1 Detailed description of communication Datapoints ..... 38
3.19.1.1 OnOff ..... 38
3.19.1.2 Relative Set Value ..... 38
3.19.1.3 Absolute Set RGB Value ..... 38
3.19.1.4 Absolute Set RGB Color Timed ..... 39
3.19.1.5 Relative Set Hue ..... 39
3.19.1.6 Relative Set Saturation ..... 39
3.19.1.7 Switch FadingShow ..... 39
3.19.1.8 Info OnOff ..... 39
3.19.1.9 Info FadingShow ..... 39
3.19.1.10 Absolute RGB Value ..... 39

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

3.20 Functional block “VIMAR 2.0 Rocker buttons ” ..... 39
3.21 Functional block “VIMAR 2.0 Rocker buttons”: Rocker button Configuration ..... 40
3.21.1 Detailed description of communication Datapoints ..... 40
3.21.1.1 Info OnOff ..... 40
3.21.1.2 OnOff ..... 40
3.22 Functional block “VIMAR 2.0 Rocker buttons”: Dimmer Configuration ..... 40
3.22.1 Detailed description of communication Datapoints ..... 40
3.22.1.1 Info OnOff ..... 40
3.22.1.2 OnOff ..... 40
3.22.1.3 Control Dimming ..... 41
3.23 Functional block “VIMAR 2.0 Rocker buttons”: Roller shutter configuration ..... 41
3.23.1 Detailed description of communication Datapoints ..... 41
3.23.1.1 StepStop_UpDown ..... 41
3.23.1.2 Move_UpDown ..... 41
3.24 Functional block “VIMAR 2.0 Rocker buttons”: Scenario Configuration ..... 41
3.24.1 Detailed description of communication Datapoints ..... 41
3.24.1.1 Scene 1 ..... 41
3.24.1.2 Scene 2 ..... 42
3.25 Functional block “VIMAR 2.0 Rocker buttons”: Vimar Audio Configuration ..... 42
3.25.1 Detailed description of communication Datapoints ..... 42
3.25.1.1 Info OnOff ..... 42
3.25.1.2 OnOff ..... 42
3.25.1.3 Control Volume ..... 42
3.25.1.4 Ch+/Track+ ..... 42
3.25.1.5 Sleep ..... 42
**4 Plug & Play home automation system and new actuators ..... 43**
4.1 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON) ..... 43
4.1.1 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Push button Configuration ..... 43
4.1.2 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Dimmer Configuration ..... 43
4.1.3 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Roller shutter configuration ..... 44
4.1.4 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Scenario Configuration ..... 44
4.1.5 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Advanced Configuration ..... 44
4.2 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON) ..... 45
4.2.1 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON): Push button Configuration ..... 45
4.2.2 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON): Dimmer Configuration ..... 45
4.2.3 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON): Roller shutter configuration ..... 46
4.3 Functional block FBT_ACT_RELAY ..... 47
4.4 Functional block FBT_ACT_ROLLER SHUTTER ..... 48
4.5 Functional block FBT_ACT_PUMP ..... 49
4.6 Functional block FBT_ACT_FANCOIL ..... 49
4.7 Functional block FBT_DIMMER_W ..... 50
4.8 Functional block FBT_DYNAMIC_W ..... 51
4.9 Functional block FBT_DIMMER_RGBW ..... 52
4.10 Functional block FBT_ACT_DIMMER2_RGB_W ..... 53
4.10.1 Functional block FBT_ACT_DIMMER2_RGB_W: White Control ..... 53
4.10.2 Functional block FBT_ACT_DIMMER2_RGB_W: Colour Control ..... 54

# SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to clarify in which way the VIMAR devices for controlling lights and roller shutters can be used in KNX or EIB networks (referred to generically in the document as KNX). The VIMAR devices are described in terms of their KNX network interface.

## 2. Composition of By-me devices

This chapter describes all the By-me devices used to control lights and roller shutters, specifying the FUNCTIONAL BLOCKS they are made up of and which are described in the next chapter, grouping the devices according to the various VIMAR wiring series.

### 2.1 Single push buttons, 16960, 14520 and 20520

It is equipped with 2 FB_V_BUTTONS functional blocks (Functional Block Index 1, 2) paired to the RH and LH buttons.

### 2.2 Single push button controls and actuator, 16965 and 14525 and 20525

It is equipped with 2 FB_V_BUTTONS functional blocks (Functional Block Index 1, 2) paired to the RH and LH buttons and an FB_V_ACTUATOR functional block (Functional Block Index 3).

### 2.3 Rocker button controls, 16961 and 14521 and 20521

It is equipped with 2 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2) paired to the RH and LH buttons.

### 2.4 Rocker button controls and actuator, 16966 and 14526 and 20526

It is equipped with 2 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2) paired to the RH and LH buttons and an FB_V_ACTUATOR functional block (Functional Block Index 3).

### 2.5 Rocker button controls and roller shutter actuator, 16967 and 14527 and20527

It is equipped with 2 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2) paired to the RH and LH buttons and an FB_V_ROLLER SHUTTER_ACTUATOR functional block (Functional Block Index 3).

### 2.6 Rocker button controls and actuator for slave controls, 16968 and 14528 and 20528

It is equipped with 2 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2) paired to the RH and LH buttons and an FB_V_DIMMER_ACTUATOR functional block (Functional Block Index 3).

N.B.: the datapoint DPT_Scaling, which enables the dimmer at a percentage value defined in the control, values from 0 to 100%, is implemented correctly from version 7 of the devices.

### 2.7 3-push button controls, 16980 and 14540 and 20540

It is equipped with 3 FB_V_BUTTONS functional blocks (Functional Block Index 1, 2, 3) paired to the RH and LH buttons.

### 2.8 Single push button controls and actuator, 16985 and 14545 and 20545

It is equipped with 3 FB_V_BUTTONS functional blocks (Functional Block Index 1, 2, 3) paired to the RH and LH buttons and an FB_V_ACTUATOR functional block (Functional Block Index 4).

### 2.9 3-rocker button controls, 16981 and 14541 and 20541

It is equipped with 3 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2, 3) paired to the RH and LH buttons.

### 2.10 3-Rocker button controls and actuator, 16986 and 14546 and 20546

It is equipped with 3 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2, 3) paired to the RH and LH buttons and an FB_V_ACTUATOR functional block (Functional Block Index 4); each block has 4 depths.

### 2.11 3-rocker button controls and roller shutter actuator, 16987 and 14547 and 20547

It is equipped with 3 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2, 3) paired to the RH and LH buttons and an FB_V_ROLLER SHUTTER_ACTUATOR functional block (Functional Block Index 4).

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 2.12 3-rocker button controls and actuator for slave controls, 16988 and 14548 and 20548

It is equipped with 3 FB_V_ROCKER_BUTTONS functional blocks (Functional Block Index 1, 2, 3) paired to the RH and LH buttons and an FB_V_DIMMER_ACTUATOR functional block (Functional Block Index 4).

N.B.: the datapoint DPT_Scaling, which enables the dimmer at a percentage value defined in the control, values from 0 to 100%.

### 2.13 Presence detector, 16935 and 14485 and 20485

It is equipped with 1 functional block FB_V_IR_DETECTOR (Functional Block Index 1).

### 2.14 Interfaces for traditional controls, 16955 and 14515 and 20515

It is equipped with 2 FB_V_CONTACTS_INTERFACE functional blocks (Functional Block Index 1, 2), corresponding to the two available inputs.

### 2.15 Interfaces for traditional controls 1M, 16958 and 14518 and 20518

It is equipped with 2 FB_V_CONTACTS_INTERFACE functional blocks (Functional Block Index 1, 2), corresponding to the two available inputs.

### 2.16 Actuators with relay output, 16975 and 14535 and 20535

It is equipped with 1 functional block FB_V_ACTUATOR (Functional Block Index 1).

### 2.17 Actuators with relay output 1M, 16974 and 14534 and 20534

It is equipped with 1 functional block FB_V_ACTUATOR (Functional Block Index 1).

### 2.18 DIN – Actuator with relay output, 1 relay, 01850.1

It is equipped with 1 functional block FB_V_ACTUATOR (Functional Block Index 1).

### 2.19 DIN – Actuator with relay output, 4 relay, 01851.1

It is equipped with 4 FB_V_ACTUATOR functional blocks (Functional Block Index 1, 2, 3, 4).

### 2.20 DIN – Actuator with relay output for two roller shutters, 01852

It is equipped with 2 FB_V_ROLLER SHUTTER_ACTUATOR functional blocks (Functional Block Index 1, 2)

### 2.21 DIN – Actuator with relay output for two roller shutters, 01852.2

It is equipped with 2 FB_V_ROLLER SHUTTER_SLATS_ACTUATOR functional blocks (Functional Block Index 1, 2).

### 2.22 DIN – Actuator with relay output for ballast, 01856

It is equipped with an FB_V_DIMMER_ACTUATOR functional block (Functional Block Index 1).

N.B.: the datapoint DPT_Scaling, which enables the dimmer at a percentage value defined in the control, values from 0 to 100%, is implemented correctly from version 4 of the devices.

### 2.23 DIN – MASTER dimmer, 01853

It is equipped with an FB_V_DIMMER_ACTUATOR functional block (Functional Block Index 1).

N.B.: the datapoint DPT_Scaling, which enables the dimmer at a percentage value defined in the control, values from 0 to 100%, is implemented correctly from version 4 of the devices.

### 2.24 DIN – 230V MASTER dimmer, 01870

It is equipped with an FB_ACT_DIMMER_W functional block (Functional Block Index 1).

### 2.25 DIN – By-me LED RGB power supply unit, 01877

It is equipped with an FB_ACT_DIMMER_RGB functional block (Functional Block Index 1).

### 2.26 DIN – Two-rocker button controls and SLAVE actuator, 16529,14529 and 20529

It is equipped with two FB_BUTTON functional blocks (Functional Block Index 1, 2) paired to the RH and LH buttons and an FB_ACT_DIMMER_RGB_W unit.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 2.27 DIN – Three-rocker button controls and MASTER actuator, 16529, 14529 and 20529

It is equipped with three FB_BUTTON functional blocks (Functional Block Index 1, 2, 3) paired to the RH, central and LH buttons and an FB_ACT_DIMMER _W block.

### 2.28 1-10Vdc actuator LED 120-230V MARINE, 01975

It is equipped with an FB_ACT_DIMMER_W functional block (Functional Block Index 1).

### 2.29 1-10Vdc actuator LED 12-24V MARINE, 01976

It is equipped with an FB_V_DIMMER_ACTUATOR functional block (Functional Block Index 1).

### 2.30 PWM actuator for 12-24V LED relay MARINE, 01978

It is equipped with an FB_V_DIMMER_ACTUATOR functional block (Functional Block Index 1).

### 2.31 Four-button home automation control, 21520

It is equipped with two FB_BUTTON functional blocks (Functional Block Index 1, 2) paired to the LH and RH button.

### 2.32 Four-button home automation control, 21520.1

It is equipped with four FBT_BUTTON functional blocks (Functional Block Index 1-4) and two FBT_ROCKER BUTTON functional blocks (Functional Block Index 5 and 6).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.33 6-button home automation control, 21540

It is equipped with three FB_BUTTON functional blocks (Functional Block Index 1, 2, 3) paired to the LH, central and LH buttons.

### 2.34 6-button home automation control, 21540.1

It is equipped with six FBT_BUTTON functional blocks (Functional Block Index 1-6) and three FBT_ROCKER BUTTON functional blocks (Functional Block Index 7-9).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.35 Multi-function home automation module 9in 8out relay, 01470

It is equipped with nine FBT_BUTTON functional blocks (Functional Block Index 1-9), four FBT_ROCKER BUTTON functional blocks (Functional Block Index 10-13), eight FBT_ACT_RELAY functional blocks (Functional Block Index 14-21), four FBT_ACT_ROLLER SHUTTER functional blocks (Functional Block Index 22-25) and an FBT_ACT_PUMP functional block (Functional Block Index 26).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.36 3in 3out LED home automation module, 01475

It is equipped with three FBT_BUTTON functional blocks (Functional Block Index 1-3) and an FBT_ROCKER BUTTON functional block (Functional Block Index 4).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.37 2in 3out home automation module (1 roll-shut+2 LED), 01476

It is equipped with two FBT_BUTTON functional blocks (Functional Block Index 1-2), an FBT_ROCKER BUTTON functional block (Functional Block Index 3) and an FBT_ACT_ROLLER SHUTTER functional block (Functional Block Index 4).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 2.38 2in 3out home automation module (1 relay+2 LEDs), 01477

It is equipped with two FBT_BUTTON functional blocks (Functional Block Index 1-2), an FBT_ROCKER BUTTON functional block (Functional Block Index 3) and an FBT_ACT_RELAY functional block (Functional Block Index 4).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.39 DIN – Multifunction home automation actuator 4out relay, 01471

It is equipped with four FBT_ACT_RELAY functional blocks (Functional Block Index 1-4), two FBT_ACT_ROLLER SHUTTER functional blocks (Functional Block Index 5-6) and an FBT_ACT_FANCOIL functional block (Functional Block Index 7).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.40 2M 4-button home automation control, 01480

It is equipped with four FBT_BUTTON functional blocks (Functional Block Index 1-4) and two FBT_ROCKER BUTTON functional blocks (Functional Block Index 5 and 6).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.41 2M 4-button home automation control+actuator, 01481

It is equipped with four FBT_BUTTON functional blocks (Functional Block Index 1-4), two FBT_ROCKER BUTTON (Functional Block Index 5 and 6) and an FBT_ACT_RELAY functional block (Functional Block Index 7).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.42 2M 4-button home automation control+roller shutter+slats, 01482

It is equipped with four FBT_BUTTON functional blocks (Functional Block Index 1-4), two FBT_ROCKER BUTTON functional blocks (Functional Block Index 5 and 6) and an FBT_ACT_ROLLER SHUTTER functional block (Functional Block Index 7).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.43 3M 6-button home automation control, 01485

It is equipped with six FBT_BUTTON functional blocks (Functional Block Index 1-6) and three FBT_ROCKER BUTTON functional blocks (Functional Block Index 7-9).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.44 3M 6-button home automation control+actuator, 01486

It is equipped with six FBT_BUTTON functional blocks (Functional Block Index 1-6), two FBT_ROCKER BUTTON functional blocks (Functional Block Index 7-9) and an FBT_ACT_RELAY functional block (Functional Block Index 10).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.45 3M 6-button home automation control+roller shutter+slats, 01487

It is equipped with six FBT_BUTTON functional blocks (Functional Block Index 1-6), two FBT_ROCKER BUTTON functional blocks (Functional Block Index 7-9) and an FBT_ACT_ROLLER SHUTTER functional block (Functional Block Index 10).

NOTE: Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.46 Home automation actuator+RGBW 4OUT controller, 01417

It is equipped with five FBT_DIMMER_W functional blocks (Functional Block Index 1-5), two FBT_DYNAMIC_W functional blocks (Functional Block Index 6-7) and an FBT_DIMMER_RGBW functional block (Functional Block Index 8).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 2.47 Home automation controller 2OUT200W LED120-240V, 01418

It is equipped with three FBT_DIMMER_W functional blocks (Functional Block Index 1-3).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.48 4-button home automation control+ LED controller 240V 2M, 01488

It is equipped with four FBT_BUTTON functional blocks (Functional Block Index 1-4), two FBT_ROCKER BUTTON functional blocks (Functional Block Index 5 and 6) and an FBT_ACT_DIMMER2_RGB_W functional block (Functional Block Index 7).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

### 2.49 4-button home automation control+controller 0/1-10V 2M, 01489

It is equipped with four FBT_BUTTON functional blocks (Functional Block Index 1-4), two FBT_ROCKER BUTTON functional blocks (Functional Block Index 5 and 6) and an FBT_ACT_DIMMER_W functional block (Functional Block Index 7).

**NOTE:** Some functional blocks reside on the same hardware portion and are therefore considered incompatible. Consult the device manual for more information.

## 3. Description of By-me device functional blocks

This chapter describes the common features of the By-me devices in order to offer a systematic vision and facilitate reading.

For a full description of the VIMAR devices, refer to the respective instruction manuals.

To facilitate understanding, here we define some VIMAR FUNCTIONAL BLOCKS that are used in the various devices listed in the following chapter. The functional blocks describe one or more functions which cannot be separated and which are together in the same device. VIMAR functional blocks are not blocks defined by the KNX standard.

Each By-me functional block is equipped with different input and output communication OBJECTS. Each single OBJECT can be connected to a GROUP ADDRESS, and can therefore be connected with other devices in the KNX network

The OBJECTS defined in these VIMAR FUNCTIONAL BLOCKS may have INPUT OBJECT and OUTPUT OBJECT functions, or both INPUT and OUTPUT functions at the same time, using the same GROUP ADDRESS value.

### 3.1 Functional block “VIMAR Rocker Buttons”

Name: FB_V_ROCKER_BUTTONS

This functional block is special as it has different group objects depending on the chosen configuration. To facilitate the description of the By-me devices it is described as a single functional block.

This functional block is the “father” of other derived functional blocks, for example single push buttons.

*Functional description*

The rocker button has two active positions, the upper and the lower part of the button. This functional block implements the push button function in different operating modes depending on the parameter configuration.

*Application*

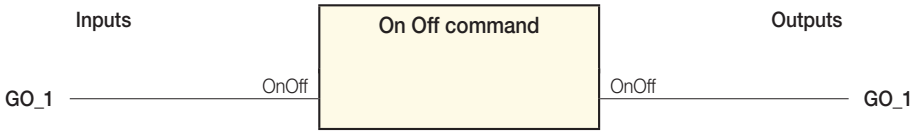
The devices that implement this functional block can be used to switch lights on and off, control dimmers, control roller shutter actuators and call up scenarios.

*N.B.*

FB_V_ROCKER_BUTTONS presents 4 different group object configurations depending on how it is configured through the By-me control panel. Here below the four different modes are described: OnOff, Dimmer control, Roller shutter control and Scenario control.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.2 Functional block “VIMAR Rocker Buttons”: On Off control configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| | **GO_1** (OnOff) | IN |
| **DPTx_OnOff *** | **GO_1** (OnOff) | OUT |

#### 3.2.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.2.1.1 OnOff (output)

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: T

*Description*: Value sent when the button is pressed, On when pressed at the top, Off when pressed at the bottom.

##### 3.2.1.2 OnOff (input)

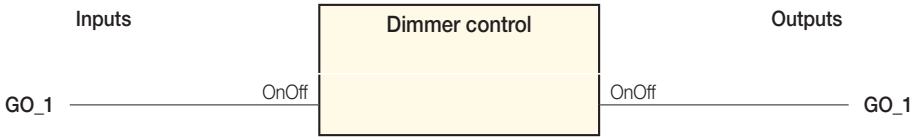
- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: According to the input values and according to the configuration, the device LEDs come on; see reference manual.

| Configuration from VIEW Pro: On - Off button | |
|---|---|
| Operation | BUS |
| Each time the button is pressed, the ON value is sent if pressed at the top, the OFF value if pressed at the bottom. | On = BC s1 s2 G1 G2 E1 00 81<br>Off = BC s1 s2 G1 G2 E1 00 80 |

To calculate the value to enter as a group address we need to divide the hex value into two octets, which represent the most and least significant parts of the value. For example, 0B0A is composed of a most significant part, 0D, and a least significant part, 0A.

### 3.3 Functional block “VIMAR Rocker Buttons”: Dimmer control configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| | **GO_1** (OnOff) | IN |
| **DPTx_OnOff *** | **GO_1** (OnOff) | OUT |
| **DPTx_DimmingControl *** | **GO_2** (Dimming Ctrl) | OUT |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.3.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.3.1.1 OnOff (output)

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: T

*Description*: Value sent when the button is pressed briefly, On when pressed at the top, Off when pressed at the bottom.

#### 3.3.1.2 OnOff (input)

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: According to the input values and according to the configuration, the device LEDs come on; see reference manual.

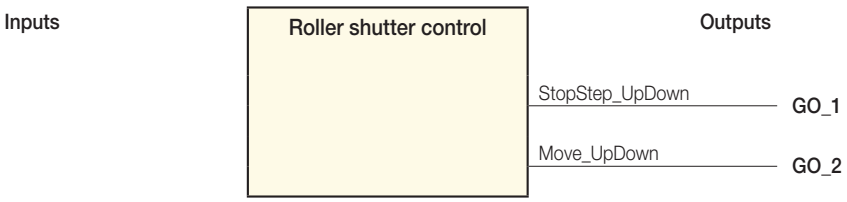
#### 3.3.1.3 Dimming Ctrl

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 2
- Flags: T

*Description*: Value sent when the button is held down: increase when pressed at the top, decrease when pressed at the bottom.

| Configuration from VIEW Pro: Dimmer control button | |
|---|---|
| Operation | BUS |
| Short press at the top: On (release, no effect)<br>Short press at the bottom: Off (release, no effect) | On = BC s1 s2 G1 G2 E1 00 81<br>Off = BC s1 s2 G1 G2 E1 00 80 |
| Long press at the top: start light increase in timed steps.<br>Release (after long press at the top): stop increase | Step+ = BC s1 s2 G1 G2 E1 00 89<br>Stop+ = BC s1 s2 G1 G2 E1 00 88 |
| Long press at the bottom: start light decrease in timed steps.<br>Release (after long press at the bottom): stop decrease. | Step- = BC s1 s2 G1 G2 E1 00 81<br>Stop- = BC s1 s2 G1 G2 E1 00 80 |

### 3.4 Functional block “VIMAR Rocker Buttons”: Roller shutter control configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_StopStepUpDown** | **GO_1** (StopStep_UpDown) | OUT |
| **DPTx_UpDown** | **GO_2** (Move_UpDown) | OUT |

### 3.4.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.4.1.1 StopStep_UpDown

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 1
- Flags: T

*Description*: Value sent when the button is pressed briefly, stops the movement when pressed at the top or the bottom.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

#### 3.4.1.2 Move_UpDown

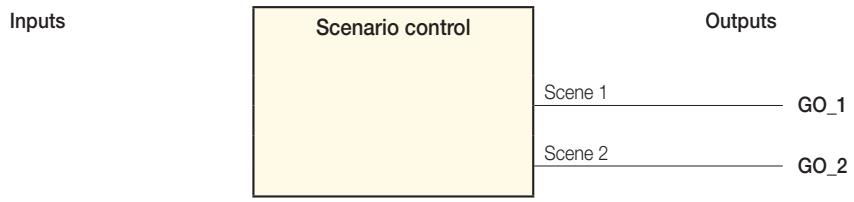
- Datapoint type: DPT_UpDown (1.008)
- Communication method: Group Object Number 2
- Flags: T

*Description*: Value sent when the button is held down: raised when pressed at the top, lowered when pressed at the bottom.

| Configuration from VIEW Pro: Roller shutter control button | |
|---|---|
| Operation | BUS |
| Short press at the top: Stop (release, no effect)<br>Short press at the bottom: Stop (release, no effect) | Stop = BC s1 s2 G1 G2 E1 00 81<br>Stop = BC s1 s2 G1 G2 E1 00 80 |
| Long press at the top: start roller shutter up, when released no effect.<br>Long press at the bottom: start roller shutter down, when released no effect. | Start+ = BC s1 s2 G1 G2 E1 00 80<br>Start- = BC s1 s2 G1 G2 E1 00 81 |



### 3.5 Functional block "VIMAR Rocker Buttons": Scenario control configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_SceneActivator1** | **GO_1** (Scene 1) | OUT |
| **DPTx_SceneActivator2** | **GO_2** (Scene 2) | OUT |

#### 3.5.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.5.1.1 Scene 1

- Datapoint type: VIM_DPT_SceneControl (18.001)
- Communication method: Group Object Number 1
- Flags: T

*Description*: Value sent when the top button is pressed

#### 3.5.1.2 Scene 2

- Datapoint type: VIM_DPT_SceneControl (18.001)
- Communication method: Group Object Number 2
- Flags: T

*Description*: Value sent when the bottom button is pressed

| Configuration from VIEW Pro: Scenario control button | |
|---|---|
| Operation | BUS |
| Whenever the button is pressed the command is sent with the scenario number. When released no message is sent | Sc = BC s1 s2 G1 G2 E2 00 3<br>NN = 1, 2, …, 32 |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

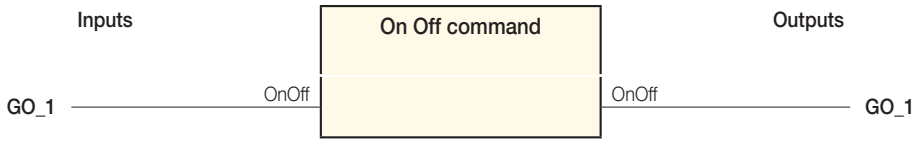
### 3.6 Functional block “VIMAR Single push buttons”

Name: FB_V_SINGLE_BUTTONS

*Functional description*

The functional block implements the push button function in different operating modes depending on the parameter configuration. The push button has only one active position, the bottom part of the button. This functional block is also special as it has different group objects depending on the chosen configuration.

### 3.7 Functional block “VIMAR Single push buttons”: On Off control configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| | **GO_1** (OnOff) | IN |
| **DPTx_OnOff *** | **GO_1** (OnOff) | OUT |

#### 3.7.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.7.1.1 OnOff (output)

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: T

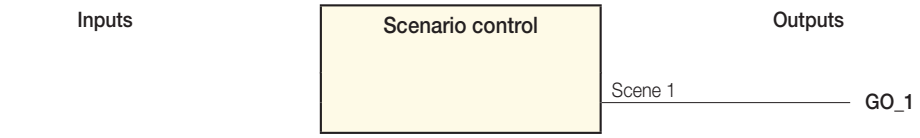
*Description*: Value sent when the button is pressed, On when pressed at the top, Off when pressed at the bottom.

##### 3.7.1.2 OnOff (input)

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: According to the input values and according to the configuration, the device LEDs come on; see reference manual.

### 3.8 Functional block “VIMAR Single push buttons”: Scenario control configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_SceneActivator1** | **GO_1** (Scene 1) | OUT |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.8.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.8.1.1 Scene 1

- Datapoint type: VIM_DPT_SceneControl (18.001)
- Communication method: Group Object Number 1
- Flags: T

*Description*: Value sent when the top button is pressed

Table and description of the button functions with the correlation between the functions and the functional blocks defined above. The functions are configured using configuration parameters in the VIEW Pro app.

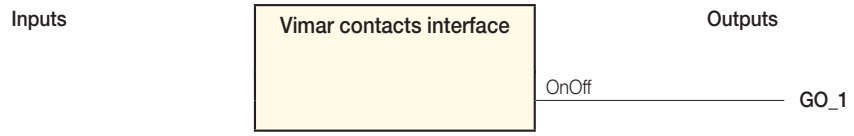
| Type of button operation | Operation | Description of functional block, see FB_V_ROCKER_BUTTONS mode: |
|---|---|---|
| Generic button | When the button is pressed the value ON is sent. When the button is released the value OFF is sent | **OnOff control**<br>On = BC s1 s2 G1 G2 E1 00 81<br>Off = BC s1 s2 G1 G2 E1 00 80 |
| Toggle button | When the button is pressed it sends alternatively the value ON or OFF. When released no message is sent | **OnOff control**<br>On = BC s1 s2 G1 G2 E1 00 81<br>Off = BC s1 s2 G1 G2 E1 00 80 |
| On only button | When the button is pressed only and always the value ON is sent | **OnOff control**<br>On = BC s1 s2 G1 G2 E1 00 81 |
| Off only button | When the button is pressed only and always the value OFF is sent | **OnOff control**<br>Off = BC s1 s2 G1 G2 E1 00 80 |
| Scenario button | Whenever the button is pressed the command is sent with the scenario number. When released no message is sent | **Scenario control**<br>Sc = BC s1 s2 G1 G2 E2 00 81 NN<br>NN = 1, 2, …, 32 |

### 3.9 Functional block “VIMAR contact interface”

Name: FB_V_CONTACTS_INTERFACE

*Functional description*

This functional block implements the interface function with two input contacts in different output operating modes depending on the parameter configuration.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff *** | **GO_1** (OnOff) | OUT |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.9.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.9.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: T

*Description*: Value sent when the contact is opened or closed, see operation table for reference.

Button operation table and correlation with the functions. The functions are configured using configuration parameters in the VIEW Pro app.

| Num | Configuration from VIEW Pro | Operation | BUS |
|---|---|---|---|
| 1 | Normal | When the contact is closed the ON message is sent, when it is opened the OFF message is sent | Close = BC s1 s2 G1 G2 E1 00 81<br>Open = BC s1 s2 G1 G2 E1 00 80 |
| 2 | Reversed | When the contact is closed the OFF message is sent, when it is opened the ON message is sent | Close = BC s1 s2 G1 G2 E1 00 80<br>Open = BC s1 s2 G1 G2 E1 00 81 |
| 3 | Toggle on rising edge | The message is sent when the contact is closed. The message is ON if the previous status was OFF, or OFF if the previous status was ON. | Msg = BC s1 s2 G1 G2 E1 00 80/1 |
| 4 | Toggle on falling edge | The message is sent when the contact is opened. The message is ON if the previous status was OFF, or OFF if the previous status was ON. | Msg = BC s1 s2 G1 G2 E1 00 80/1 |

*Parameters*

The parameters can be edited in VIEW Pro.

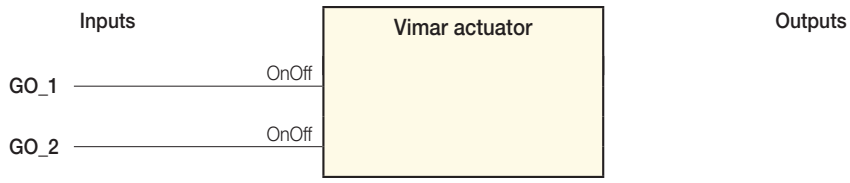
| Field | Values |
|---|---|
| Operation type | Normal, reversed, Toggle on rising edge, toggle on falling edge. |
| LED management | LED Disabled (0), Enabled (1) |

### 3.10 Functional block “VIMAR actuator”

Name: FB_V_ACTUATOR

*Functional description*

This functional block implements the relay actuator function in different operating modes depending on the parameter configuration. See table below.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff *** | **GO_1** (OnOff) | IN |
| **DPTx_OnOff *** | **GO_2** (OnOff) | IN |

### 3.10.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.10.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1 and 2
- Flags: W

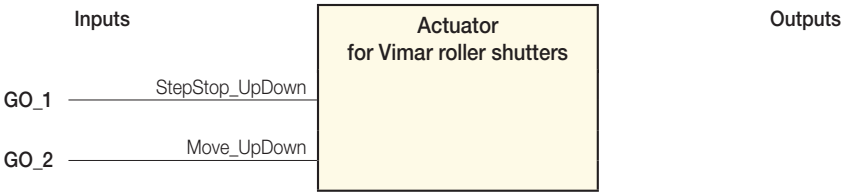
*Description*: input to activate the actuator or not, see actuator operating parameters table for reference.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.11 Functional block “VIMAR Roller shutter actuator”

Name: FB_V_ROLLER SHUTTER_ACTUATOR

*Functional description*
This functional block implements the relay actuator function to control a roller shutter in different operating modes depending on the parameter configuration. See table below.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_StopStepUpDown** | **GO_1** (StopStep_UpDown) | IN |
| **DPTx_UpDown** | **GO_2** (Move_UpDown) | IN |

#### 3.11.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.11.1.1 StepStop_UpDown

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Control to stop the output, whether the shutter is going up or down

##### 3.11.1.2 Move_UpDown

- Datapoint type: DPT_UpDown (1.008)
- Communication method: Group Object Number 2
- Flags: W

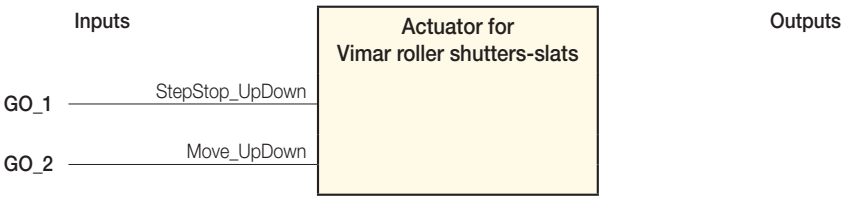
*Description*: Control to raise or lower the roller shutter, see actuator operating parameters table for reference.

*N.B.*
When configuring the integrated device in a KNX network it is important to remember that two objects must be configured to ensure the correct operation of the roller shutter actuator: StopStep_UpDown and Move_UpDown. The object StopStep_UpDown is used to stop the roller shutter movement by switching off the roller shutter, while the object Move_UpDown is used to move the shutter up and down.

### 3.12 Functional block “VIMAR Roller shutter-Slats actuator”

Name: FB_V_ROLLER SHUTTER_SLATS_ACTUATOR

*Functional description*
This functional block implements the relay actuator function to control a roller shutter and/or slats in different operating modes depending on the parameter configuration. See table below.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_StopStepUpDown** | **GO_1** (StopStep_UpDown) | IN |
| **DPTx_UpDown** | **GO_2** (Move_UpDown) | IN |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.12.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.12.1.1 StopStep_UpDown

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Control to stop the output, whether the shutter is going up or down. In slats operation, the slats move up or down.

*Normal mode:*
0/1: STOP. The message is received to switch off the channel regardless of the value received.

*Slats mode:*

Roller shutter moving:

0/1: STOP. The message is received to switch off the channel regardless of the value received.

Roller shutter stopped:

0: STEP DOWN. Slats move down.

1: STEP UP. Slats move up.

#### 3.12.1.2 Move_UpDown

- Datapoint type: DPT_UpDown (1.008)
- Communication method: Group Object Number 2
- Flags: W

*Description*: Control to raise or lower the roller shutter, see actuator operating parameters table for reference.

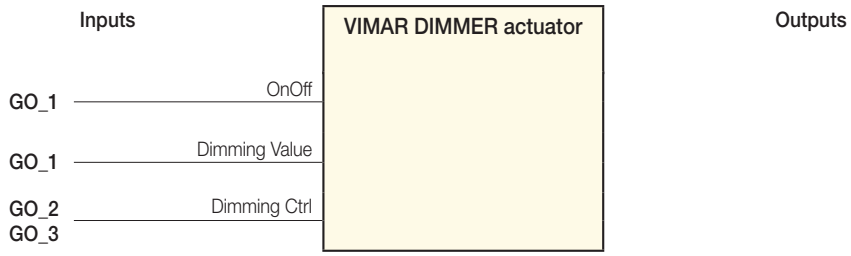
*N.B.*
When configuring the integrated device in a KNX network it is important to remember that two objects must be configured to ensure the correct operation of the roller shutter actuator: StopStep_UpDown and Move_UpDown. The object StopStep_UpDown is used to stop the roller shutter movement by switching off the roller shutter, while the object Move_UpDown is used to move the shutter up and down.

### 3.13 Functional block "VIMAR DIMMER actuator"

Name: FB_V_DIMMER_ACTUATOR

*Functional description*

This functional block controls a dimmer. We can adjust the control speed by configuring the parameters. See table below.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff *** | **GO_1** (OnOff) | IN |
| **DPTx_DimmingValue** | **GO_1** (Dimming Value) | IN |
| **DPTx_DimmingControl** | **GO_2 GO_3** (Dimming Ctrl) | IN |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

#### 3.13.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.13.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Command that enables or disables the output: the output is activated on ON and deactivated on Off. See the actuator operating parameters table for reference.

##### 3.13.1.2 Dimming Value

- Datapoint type: DPT_Scaling (5.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Command that switches on at a percentage value defined in the control, values from 0 to 100%.

##### 3.13.1.3 Dimming Ctrl

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 2 and 3
- Flags: W

*Description*: Command to activate the output or not, see actuator operating parameters table for reference. The output is activated on ON and disabled on Off.

*N.B.*
When configuring the integrated device in a KNX network it is important to remember that two objects must be configured to ensure the correct operation of the actuator: OnOff and Dimming Ctrl. The object OnOff is used to switch the actuator on and off, while the object Dimming Ctrl is used to control the intensity.



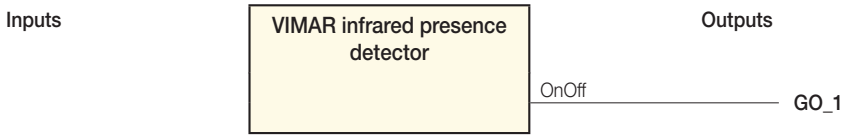
### 3.14 Functional block “VIMAR Infrared presence detector”

Name: FB_V_IR_DETECTOR

*Functional description*

This functional block implements the presence detector function for lighting in different output operating modes depending on the parameter configuration.

This device can only be configured to switch the lights on.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff** | **GO_1** (OnOff) | OUT |

#### 3.14.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.14.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: RT

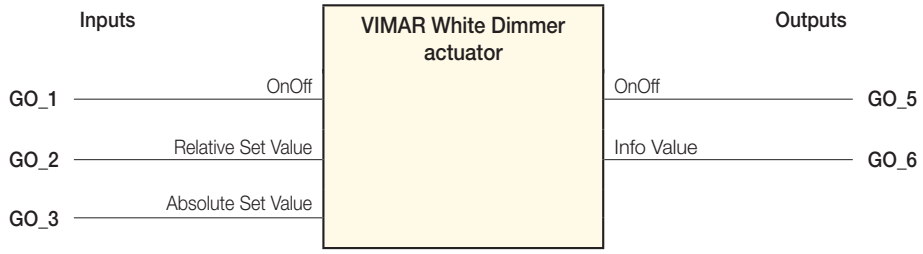
*Description*: Value sent to the presence detector.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.15 Functional block “VIMAR White Dimmer Actuator”

Name: FB_ACT_DIMMER_W

*Functional description*
This functional block controls the lights. You can change the control by configuring its parameters.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff** | **GO_1** (OnOff) | IN |
| **DPTx_DimmingControl** | **GO_2** (Relative Set Value) | IN |
| **DPTx_Brightness** | **GO_3** (Absolute Set Value) | IN |
| **DPTx_OnOffInfo** | **GO_5** (Info OnOff) | OUT |
| **DPTx_BrightnessInfo** | **GO_6** (Info Value) | OUT |

#### 3.15.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.15.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Command that enables or disables the output: the output is activated with ON and deactivated with Off.

##### 3.15.1.2 Relative Set Value

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 2
- Flags: W

*Description*: Controls activation of the output, it is activated with ON and deactivated with OFF.

##### 3.15.1.3 Absolute Set Value

- Datapoint type: DPT_Scaling (5.001)
- Communication method: Group Object Number 3
- Flags: W

*Description*: Command that switches on at a percentage value defined in the control, values vary from 0 to 100%.

##### 3.15.1.4 Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 5
- Flags: RT

*Description*: Value sent after the actuator status changes. Informs if the status is ON or OFF.

##### 3.15.1.5 Info Value

- Datapoint type: DPT_Scaling (5.001)
- Communication method: Group Object Number 6
- Flags: R

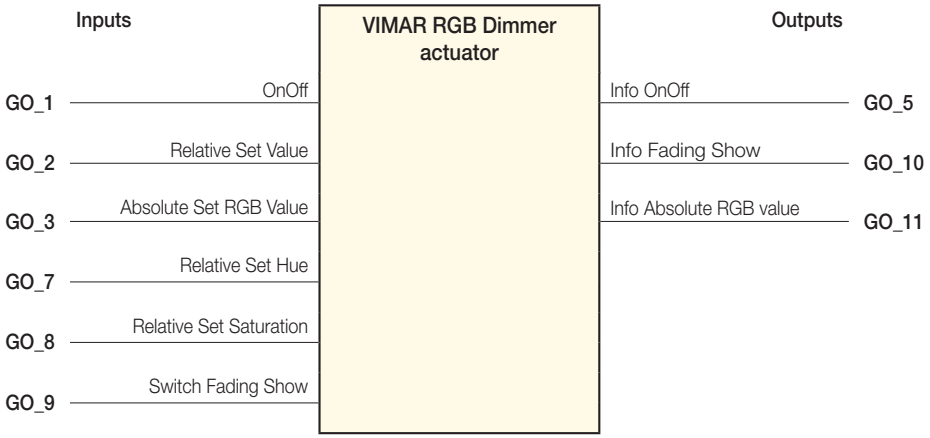
*Description*: Value sent after the actuator status changes. It informs of the percentage value of the actuator.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.16 Functional block “VIMAR RGB Dimmer Actuator”

Name: FB_ACT_DIMMER_RGB

*Functional description*
This functional block controls the lights. You can change the control by configuring its parameters.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff** | **GO_1** (OnOff) | IN |
| **DPTx_DimmingControl** | **GO_2** (Relative Set Value) | IN |
| **DPTx_RgbColour** | **GO_3** (Absolute Set RGB Value) | IN |
| **DPTx_TimedRgbColour** | **GO_6** (Absolute Set RGB Timed) | IN |
| **DPTx_ControlHue** | **GO_7** (Relative Set Hue) | IN |
| **DPTx_ControlSaturation** | **GO_8** (Relative Set Saturation) | IN |
| **DPTx_FadingShow** | **GO_9** (Switch Fading Show) | IN |
| **DPTx_OnOffInfo** | **GO_5** (Info OnOff) | OUT |
| **DPTx_InfoAbsoluteRGBColor** | **GO_11** (Info Absolute RGB value) | OUT |
| **DPTx_FadingShowInfo** | **GO_10** (Info Fading Show) | OUT |

#### 3.16.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.16.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Command that enables or disables the output: the output is activated with ON and deactivated with Off.

##### 3.16.1.2 Relative Set Value

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 2
- Flags: W

*Description*: Controls activation of the output, it is activated with ON and deactivated with OFF.

##### 3.16.1.3 Absolute Set RGB Value

- Datapoint type: DPT_Colour_RGB (232.600)
- Communication method: Group Object Number 3
- Flags: W

*Description*: Absolute value of the RGB coordinate to be set on the actuator.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

#### 3.16.1.4 Relative Set Hue

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 7
- Flags: W

*Description*: Request to increase or decrease the HUE coordinate.

#### 3.16.1.5 Relative Set Saturation

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 8
- Flags: W

*Description*: Activation of the FadingShow procedure.

#### 3.16.1.6 Switch FadingShow

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 9
- Flags: W

*Description*: Activation of the FadingShow procedure.

#### 3.16.1.7 Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 5
- Flags: RT

*Description*: Value sent after the actuator status changes. Informs if the status is ON or OFF.

#### 3.16.1.8 Info FadingShow

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 10
- Flags: RT

*Description*: Current FadingShow value.

#### 3.16.1.9 Info Absolute RGB Value

- Datapoint type: DPT_Colour_RGB (232.600)
- Communication method: Group Object Number 11
- Flags: R

*Description*: Absolute value of the RGB coordinate that is active in the actuator. Caution: this datapoint is read only

### 3.17 Functional block “VIMAR RGB and White Dimmer Actuator”:

Name: Name: FB_ACT_DIMMER_RGB_W

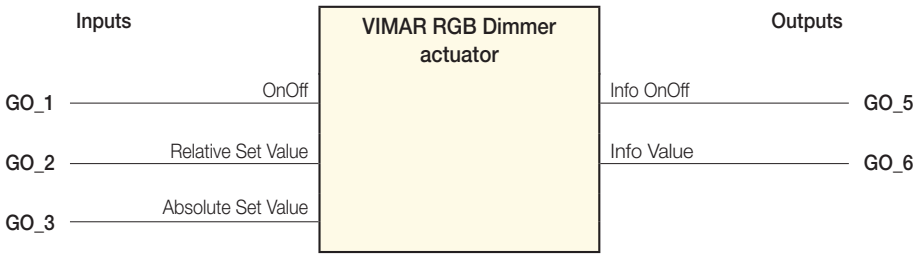
*Functional description*
This functional block controls the lights. You can change the control by configuring its parameters.

*Application*
The devices implementing this functional block can be used to increase or lower the brightness.

*N.B.*
FB_ACT_DIMMER_ RGB_W presents 2 different group object configurations depending on how it is configured through the By-me control panel. Here below the two different modes are described: White Control and Colour Control.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.18 Functional block “VIMAR RGB and White Dimmer Actuator”: White Control Configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff** | **GO_1** (OnOff) | IN |
| **DPTx_DimmingControl** | **GO_2** (Relative Set Value) | IN |
| **DPTx_Brightness** | **GO_3** (Absolute Set Value) | IN |
| **DPTx_OnOffInfo** | **GO_5** (Info OnOff) | OUT |
| **DPTx_BrightnessInfo** | **GO_6** (Info Value) | OUT |

#### 3.18.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.18.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Command that enables or disables the output: the output is activated with ON and deactivated with Off.

##### 3.18.1.2 Relative Set Value

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 2
- Flags: W

*Description*: Controls activation of the output, it is activated with ON and deactivated with OFF.

##### 3.18.1.3 Absolute Set Value

- Datapoint type: DPT_Scaling (5.001)
- Communication method: Group Object Number 3
- Flags: W

*Description*: Command that switches on at a percentage value defined in the control, values vary from 0 to 100%.

##### 3.18.1.4 Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 5
- Flags: RT

*Description*: Value sent after the actuator status changes. Informs if the status is ON or OFF.

##### 3.18.1.5 Info Absolute Value

- Datapoint type DPT_Switch (1.001)
- Communication method: Group Object Number 6
- Flags: RT

*Description*: Value sent after the actuator status changes. It informs of the percentage value of the actuator.

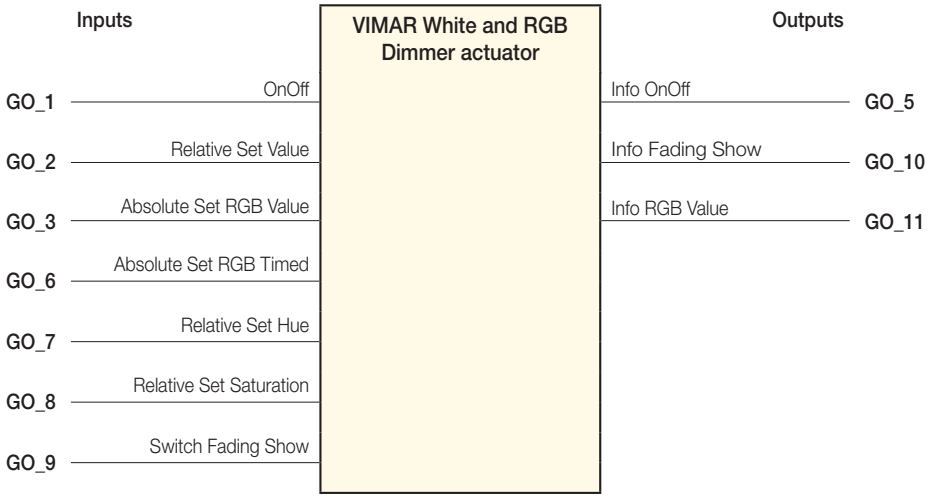
## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.19 Functional block “VIMAR RGB and White Dimmer Actuator”: Colour Control Configuration

Name: FB_ACT_DIMMER_RGB

*Functional description*

This functional block controls the lights. You can change the control by configuring its parameters.




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOff** | **GO_1** (OnOff) | IN |
| **DPTx_DimmingControl** | **GO_2** (Relative Set Value) | IN |
| **DPTx_RgbColour** | **GO_3** (Absolute Set RGB Value) | IN |
| **DPTx_TimedRgbColour** | **GO_6** (Absolute Set RGB Timed) | IN |
| **DPTx_ControlHue** | **GO_7** (Relative Set Hue) | IN |
| **DPTx_ControlSaturation** | **GO_8** (Relative Set Saturation) | IN |
| **DPTx_FadingShow** | **GO_9** (Switch Fading Show) | IN |
| **DPTx_OnOffInfo** | **GO_5** (Info OnOff) | OUT |
| **DPTx_InfoAbsoluteRGBColor** | **GO_11** (Info Absolute RGB value) | OUT |
| **DPTx_FadingShowInfo** | **GO_10** (Info Fading Show) | OUT |

#### 3.19.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.19.1.1 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Command that enables or disables the output: the output is activated with ON and deactivated with Off.

##### 3.19.1.2 Relative Set Value

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 2
- Flags: W

*Description*: Controls activation of the output, it is activated with ON and deactivated with OFF.

##### 3.19.1.3 Absolute Set RGB Value

- Datapoint type: DPT_Colour_RGB (232.600)
- Communication method: Group Object Number 3
- Flags: W

*Description*: Absolute value of the RGB coordinate to be set on the actuator. With the READ request it sends the RGB colour present in the actuator.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

#### 3.19.1.4 Absolute Set RGB Colour Timed

- Datapoint type: Custom
- Communication method: Group Object Number 6
- Flags: W

*Description*: Value in RGB coordinates to be set in the actuator within a certain time.

#### 3.29.1.5 Relative Set Hue

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 7
- Flags: W

*Description*: Request to increase or decrease the HUE (colour) coordinate.

#### 3.19.1.6 Relative Set Saturation

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 8
- Flags: W

*Description*: Request to increase or decrease the colour saturation.

#### 3.19.1.7 Switch FadingShow

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 9
- Flags: W

*Description*: Activation of the FadingShow procedure.

#### 3.19.1.8 Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 5
- Flags: RT

*Description*: Value sent after the actuator status changes. Informs if the status is ON or OFF.

#### 3.19.1.9 Info FadingShow

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 10
- Flags: RT

*Description*: Current FadingShow value.

#### 3.19.1.10 Absolute RGB Value

- Datapoint type: DPT_Colour_RGB (232.600)
- Communication method: Group Object Number 11
- Flags: R

*Description*: With the READ request it sends the RGB colour present in the actuator. Caution: the datapoint is read only.

### 3.20 Functional block “VIMAR 2.0 Rocker buttons”

Name: Name: FB_BUTTON

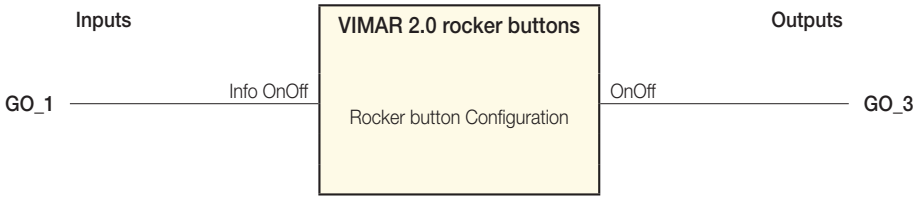
*Functional description*
The rocker button has two active positions, the upper and the lower part of the button. This functional block implements the push button function in different operating modes depending on the parameter configuration.

*Application*
The devices that implement this functional block can be used to switch lights on and off, control dimmers, control roller shutter actuators, call up scenarios and control the Vimar Audio application.

*N.B.*
FB_BUTTON presents 5 different group object configurations depending on how it is configured through the By-me control panel. Here below the four different modes are described: Rocker Control, Dimmer Control, Roller Shutter Control, Scenario Control and Vimar Audio Control.

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 3.21 Functional block “VIMAR 2.0 Rocker buttons”: Rocker button Configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOffInfo *** | **GO_1** (Info OnOff) | IN |
| **DPTx_OnOff *** | **GO_3** (OnOff) | OUT |

#### 3.21.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.21.1.1 Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

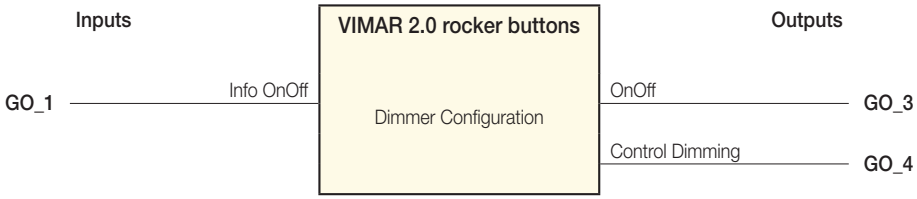
*Description*: Value sent after the actuator status changes. Informs if the actuator is ON or OFF.

##### 3.21.1.2 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 3
- Flags: RT

*Description*: Value sent when the button is pressed, On when pressed at the top, Off when pressed at the bottom.

### 3.22 Functional block “VIMAR 2.0 Rocker buttons”: Dimmer Configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOffInfo *** | **GO_1** (Info OnOff) | IN |
| **DPTx_OnOff *** | **GO_3** (OnOff) | OUT |
| **DPTx_DimmingControl *** | **GO_4** (Dimming Ctrl) | OUT |

#### 3.22.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 3.22.1.1 Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Value sent after the actuator status changes. Informs if the actuator is ON or OFF.

##### 3.22.1.2 OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 3
- Flags: RT

*Description*: Value sent when the button is pressed, On when pressed at the top, Off when pressed at the bottom.

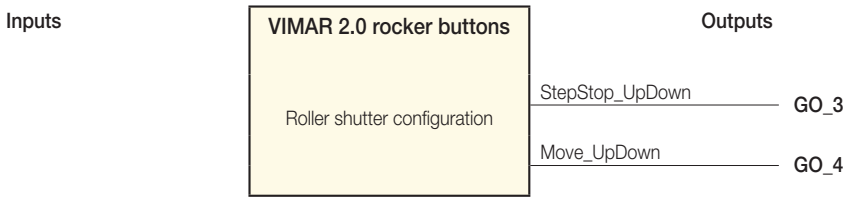
## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

#### 3.22.1.3 Control Dimming

- Datapoint type: DPT_ControlDimming (3.007)
- Communication method: Group Object Number 4
- Flags: RT

*Description*: Message used to control a dimmer actuator in the automation system.

### 3.23 Functional block "VIMAR 2.0 Rocker buttons": Roller shutter configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_StopStepUpDown** | **GO_3** (StopStep_UpDown) | OUT |
| **DPTx_UpDown** | **GO_4** (Move_UpDown) | OUT |

#### 3.23.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.23.1.1 StepStop_UpDown

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 3
- Flags: RT

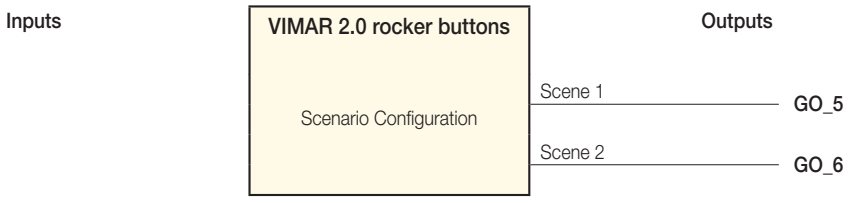
*Description*: Message sent to command the Stop of the previous actuation UP or DOWN.

#### 3.23.1.2 Move_UpDown

- Datapoint type: DPT_UpDown (1.008)
- Communication method: Group Object Number 4
- Flags: RT

*Description*: Controls the paired roller shutter actuator UP or DOWN

### 3.24 Functional block "VIMAR 2.0 Rocker buttons": Scenario Configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_SceneActivator1** | **GO_5** (Scene 1) | OUT |
| **DPTx_SceneActivator2** | **GO_6** (Scene 2) | OUT |

#### 3.24.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.24.1.1 Scene 1

- Datapoint type: DPT_SceneControl (18.001)
- Communication method: Group Object Number 5
- Flags: RT

*Description*: Message sent to control the scenario corresponding to the Down button.

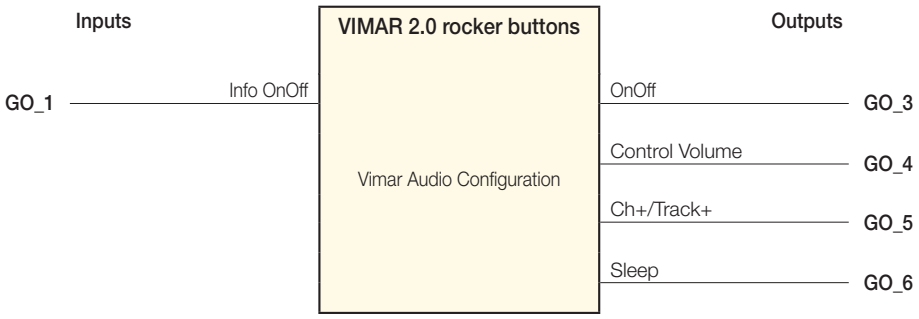
## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

#### 3.24.1.2 Scene 2

- Datapoint type: DPT_SceneControl (18.001)
- Communication method: Group Object Number 6
- Flags: RT

*Description*: Message sent to control the scenario corresponding to the Up button.

### 3.25 Functional block “VIMAR 2.0 Rocker buttons”: Vimar Audio Configuration




| Name in VIEW Pro | Datapoint | IN/OUT |
|---|---|---|
| **DPTx_OnOffInfo *** | **GO_1** (Info OnOff) | IN |
| **DPTx_OnOff *** | **GO_3** (OnOff) | OUT |
| **DPTx_VolumeControl *** | **GO_4** (Control Volume) | OUT |
| **DPTx_SkipChannelTrack *** | **GO_5** (Ch+/Track+) | OUT |
| **DPTx_Sleep *** | **GO_6** (Sleep) | OUT |

#### 3.25.1 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 3.25.1.1 Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Description*: Value sent after the actuator status changes. Informs if the actuator is ON or OFF.

#### 3.25.1.2 OnOff

- Datapoint type: DPT_Switch (1.007)
- Communication method: Group Object Number 3
- Flags: RT

*Description*: Controls the activation or deactivation of the paired actuator.

#### 3.25.1.3 Volume Control

- Datapoint type: DPT_ControlDimming (3.007)
- Communication method: Group Object Number 4
- Flags: RT

*Description*: Message sent to control increasing or decreasing the volume of the receivers of the paired audio group.

#### 3.25.1.4 Ch+/Track+

- Datapoint type: Vimar Standard
- Communication method: Group Object Number 5
- Flags: RT

*Description*: Message corresponding to the next channel (top button Ch+) and next track (bottom button Track+) functions of the paired audio receivers.

#### 3.25.1.5 Sleep

- Datapoint type: DPT_Enable (1.003)
- Communication method: Group Object Number 6
- Flags: RT

*Description*: Message to activate the Sleep function of the audio receivers of the zones paired with the button.

# SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

## 4. Plug & Play home automation system and new actuators

### 4.1 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON)
Name: FBT_BUTTON

#### Functional description

The single push button has only one active position. This functional block implements the push button function in different operating modes depending on the parameter configuration.

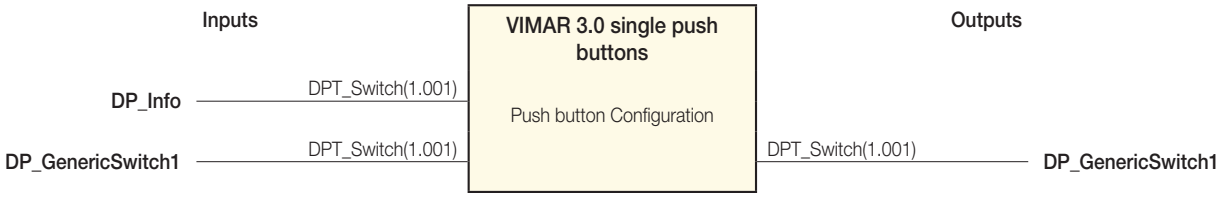
#### Application

The devices that implement this functional block can be used to switch lights on and off, control dimmers, control roller shutter actuators, call up scenarios and control timer-operated lights.

#### N.B.

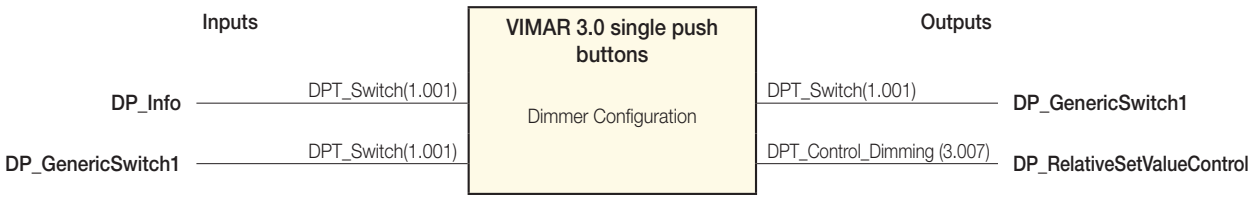
FB_BUTTON presents 5 different group object configurations depending on how it is configured in VIEW Pro. The five different modes are described below: Push Button, Dimmer Control, Roller Shutter Control, Scenario Control and Advanced Control.

#### 4.1.1 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Push button Configuration




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo *** | **DP_Info** | DPT_Switch(1.001) | 1 | W |
| Value sent after the actuator status changes. Informs if the actuator is ON or OFF. | | | | |
| **DPTx_OnOff *** | **DP_GenericSwitch1** | DPT_Switch(1.001) | 3 | RWT |
| Controls the activation or deactivation of the paired actuator. | | | | |

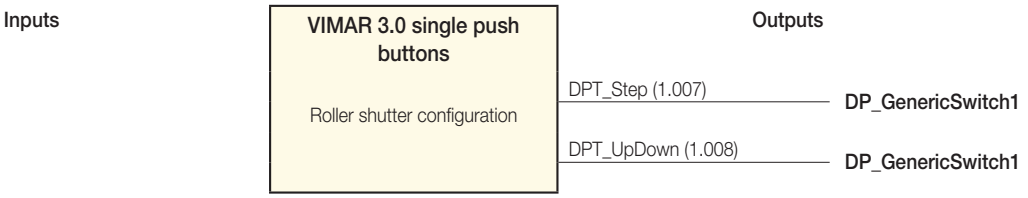
#### 4.1.2 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Dimmer Configuration




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo *** | **DP_Info** | DPT_Switch(1.001) | 1 | W |
| Value sent after the actuator status changes. Informs if the actuator is ON or OFF. | | | | |
| **DPTx_OnOff *** | **DP_GenericSwitch1** | DPT_Switch(1.001) | 3 | RWT |
| Controls the activation or deactivation of the paired actuator. | | | | |
| **DPTx_ControlDimming *** | **DP_RelativeSetValueControl** | DPT_Control_Dimming (3.007) | 5 | T |
| Message used to control a dimmer actuator in the automation system. | | | | |

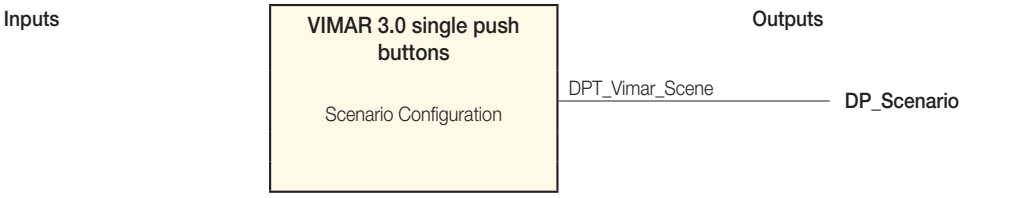
## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.1.3 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Roller shutter configuration



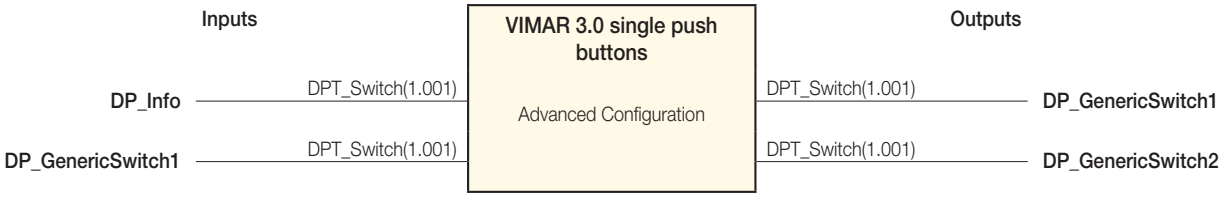

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_StopStepUpDown** | **DP_GenericSwitch1** | DPT_Step (1.007) | 3 | T |
| Message sent to command the Stop of the previous actuation UP or DOWN or, if the actuator is stopped, to move the slats. | | | | |
| **DPTx_UpDown** | **DP_GenericSwitch1** | DPT_UpDown (1.008) | 4 | T |
| Controls the paired roller shutter actuator UP or DOWN | | | | |

### 4.1.4 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Scenario Configuration




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_SceneActivator1** | **DP_Scenario** | DPT_Vimar_Scene | 7 | T |
| Message sent to control the scenario corresponding to the button. | | | | |

### 4.1.5 Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON): Advanced Configuration




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo *** | **DP_Info** | DPT_Switch(1.001) | 1 | W |
| Value sent after the actuator status changes. Informs if the actuator is ON or OFF. | | | | |
| **DPTx_OnOff *** | **DP_GenericSwitch1** | DPT_Switch(1.001) | 3 | RWT |
| Value sent when the push button is pressed briefly. | | | | |
| **DPTx_TimedStartStop *** | **DP_GenericSwitch2** | DPT_Switch(1.001) | 4 | T |
| Value sent when the push button is held down. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.2 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON)
Name: FBT_ROCKER BUTTON

#### Functional description

The rocker button has two active positions. This functional block implements the push button function in different operating modes depending on the parameter configuration.

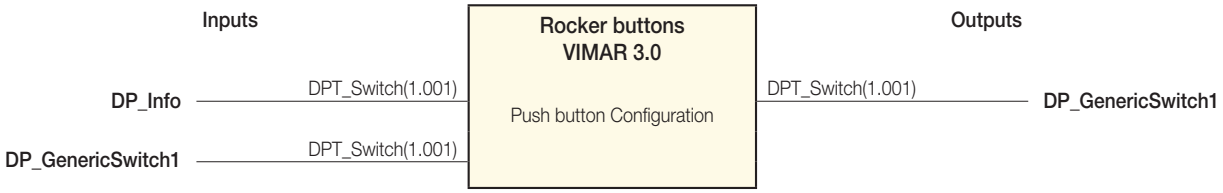
#### Application

The devices that implement this functional block can be used to switch lights on and off, control dimmers, control roller shutter actuators and control timer-operated lights.

#### N.B.

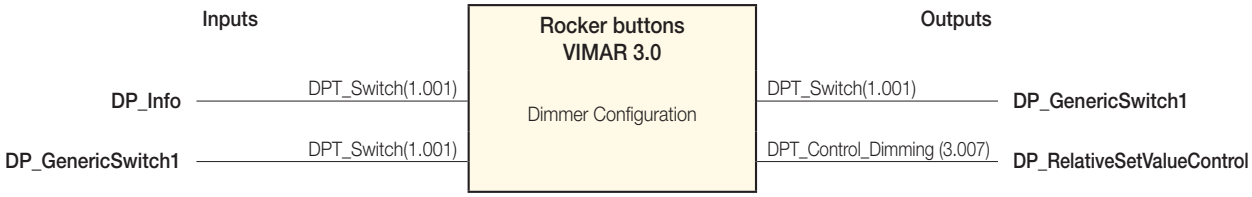
The FBT_ROCKER BUTTON presents 4 different group object configurations depending on how it is configured in VIEW Pro. Here below the four different modes are described: Push button, Dimmer control, Roller shutter control and Advanced control.

#### 4.2.1 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON): Push button Configuration




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo *** | **DP_Info** | DPT_Switch(1.001) | 1 | W |
| Value sent after the actuator status changes. Informs if the actuator is ON or OFF. | | | | |
| **DPTx_OnOff *** | **DP_GenericSwitch1** | DPT_Switch(1.001) | 3 | RWT |
| Controls the activation or deactivation of the paired actuator. | | | | |

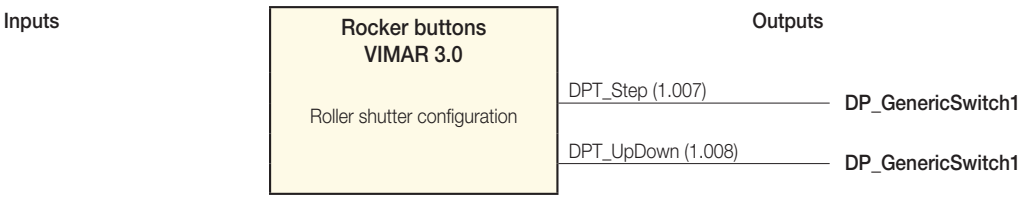
#### 4.2.2 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON): Dimmer Configuration




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo *** | **DP_Info** | DPT_Switch(1.001) | 1 | W |
| Value sent after the actuator status changes. Informs if the actuator is ON or OFF. | | | | |
| **DPTx_OnOff *** | **DP_GenericSwitch1** | DPT_Switch(1.001) | 3 | RWT |
| Controls the activation or deactivation of the paired actuator. | | | | |
| **DPTx_ControlDimming *** | **DP_RelativeSetValueControl** | DPT_Control_Dimming (3.007) | 5 | T |
| Message used to control a dimmer actuator in the automation system. | | | | |

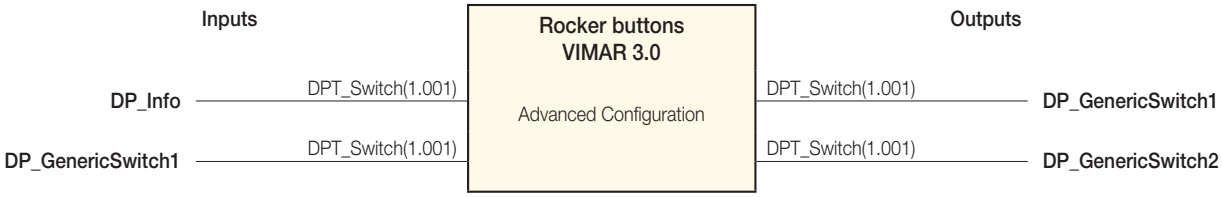
## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.2.3 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON): Roller shutter configuration




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_StopStepUpDown** | **DP_GenericSwitch1** | DPT_Step (1.007) | 3 | T |
| Message sent to command the Stop of the previous actuation UP or DOWN or, if the actuator is stopped, to move the slats. | | | | |
| **DPTx_UpDown** | **DP_GenericSwitch1** | DPT_UpDown (1.008) | 4 | T |
| Controls the paired roller shutter actuator UP or DOWN | | | | |

### 4.2.4 Functional block “VIMAR 3.0 Rocker buttons” (FBT_ROCKER BUTTON): Advanced Configuration



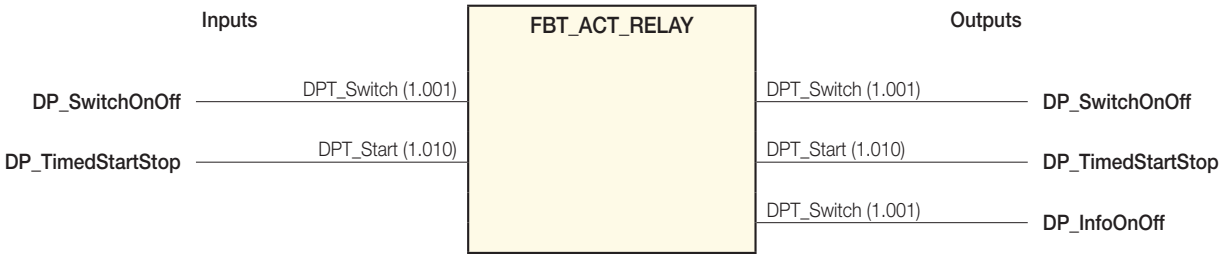

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo *** | **DP_Info** | DPT_Switch(1.001) | 1 | W |
| Value sent after the actuator status changes. Informs if the actuator is ON or OFF. | | | | |
| **DPTx_OnOff *** | **DP_GenericSwitch1** | DPT_Switch(1.001) | 3 | RWT |
| Value sent when the push button is pressed briefly. | | | | |
| **DPTx_TimedStartStop *** | **DP_GenericSwitch2** | DPT_Switch(1.001) | 4 | T |
| Value sent when the push button is held down. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.3 Functional block FBT_ACT_RELAY
Name: FBT_ACT_RELAY

#### Functional description

This functional block implements the relay actuator function in different operating modes depending on the parameter configuration.



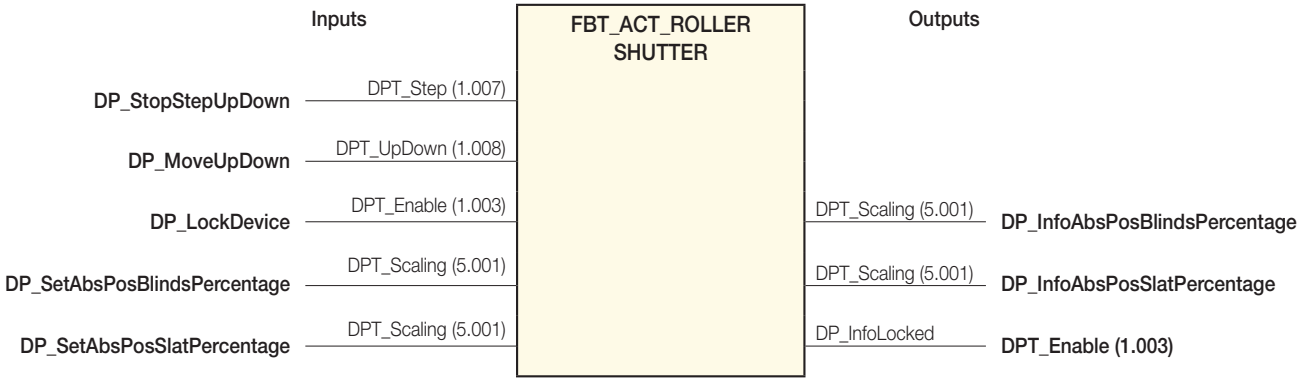

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff *** | **DP_SwitchOnOff** | DPT_Switch (1.001) | 1 | RWT |
| Controls the relay enabling in bistable mode. | | | | |
| **DPTx_TimedStartStop** | **DP_TimedStartStop** | DPT_Start (1.010) | 4 | RWT |
| Controls the relay enabling in one-position stable mode. | | | | |
| **DPTx_OnOffInfo *** | **DP_InfoOnOff** | DPT_Switch (1.001) | 6 | RT |
| Value sent after the actuator status changes. Informs if the actuator is ON or OFF. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.4 Functional block FBT_ACT_ROLLER SHUTTER
Name: FBT_ACT_ROLLER SHUTTER

#### Functional description

This functional block implements the relay actuator function to control a roller shutter in different operating modes depending on the parameter configuration.

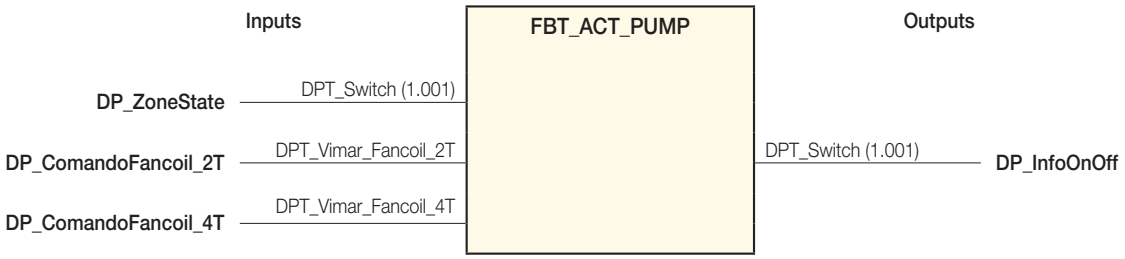



| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_StopStepUpDown** | **DP_StopStepUpDown** | DPT_Step (1.007) | 1 | W |
| If the roller shutters are moving on receipt of 0 or 1 roller shutter stop command.<br>If the roller shutters are stopped, slat rotation command (1-->open, 0-->close). | | | | |
| **DPTx_UpDown** | **DP_MoveUpDown** | DPT_UpDown (1.008) | 2 | W |
| Up_down roller shutter command (0-->open, 1-->close). | | | | |
| **DPTx_Lock** | **DP_LockDevice** | DPT_Enable (1.003) | 3 | W |
| Used for forcing with roller shutter up and slat tilt as set in the configuration with the parameters PAR_LockBehaviour, PAR_BlindsPositionForLock and PAR_SlatPositionForLock. | | | | |
| **DPTx_ShutterPosition** | **DP_SetAbsPosBlindsPercentage** | DPT_Scaling (5.001) | 4 | W |
| Receipt of percentage value of the roller shutter height<br>0% = all open<br>100% = all closed | | | | |
| **DPTx_SlatPosition **** | **DP_SetAbsPosSlatPercentage** | DPT_Scaling (5.001) | 5 | W |
| Receipt of percentage value of the roller shutter angle<br>0% = all open<br>100% = all closed | | | | |
| **DPTx_ShutterPositionInfo** | **DP_InfoAbsPosBlindsPercentage** | DPT_Scaling (5.001) | 7 | RT |
| Sending of current percentage value of the roller shutter height<br>0% = all open<br>100% = all closed | | | | |
| **DPTx_SlatPositionInfo **** | **DP_InfoAbsPosSlatPercentage** | DPT_Scaling (5.001) | 8 | RT |
| Sending of current percentage value of the roller shutter angle<br>0% = all open<br>100% = all closed | | | | |
| **DPTx_LockInfo** | **DP_InfoLocked** | DPT_Enable (1.003) | 9 | RT |
| Feedback on forcing status. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.5 Functional block FBT_ACT_PUMP
Name: FBT_ACT_PUMP

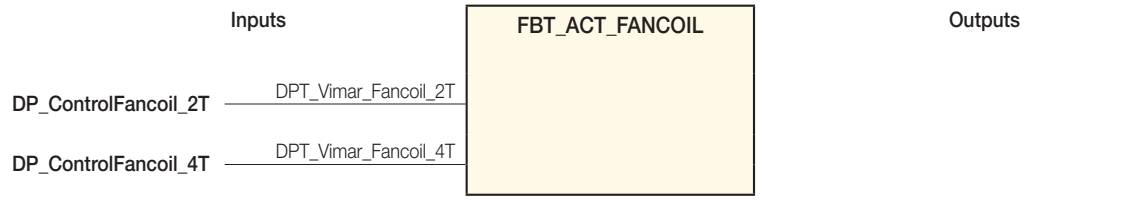
#### Functional description
This functional block controls the actuations of a climate zone.




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_ZoneState** | DPT_Switch (1.001) | 1 | W |
| Monitors up to 64 climate zones and calculates the OR to change the status of the relay connected to the circulation pump. | | | | |
| **DPTx_Fancoil2Pipes** | **DP_ComandoFancoil_2T** | DPT_Vimar_Fancoil_2T | 2 | W |
| Like DP_ZoneState but referred to 2-pipe fan coil devices. | | | | |
| **DPTx_Fancoil4Pipes** | **DP_ComandoFancoil_4T** | DPT_Vimar_Fancoil_4T | 3 | W |
| Like DP_ZoneState but referred to 4-pipe fan coil devices. | | | | |
| **DPTx_InfoOnOff** | **DP_InfoOnOff** | DPT_Switch (1.001) | 4 | RT |
| Feedback on output status. | | | | |

### 4.6 Functional block FBT_ACT_FANCOIL
Name: FBT_ACT_FANCOIL

#### Functional description
This functional block controls a fan coil actuator.



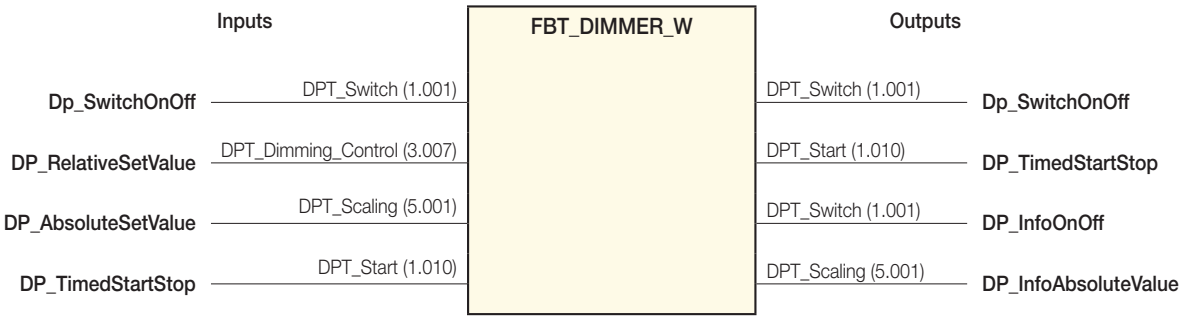

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_Fancoil2Pipes** | **DP_ControlFancoil_2T** | DPT_Vimar_Fancoil_2T | 1 | W |
| Receipt of Vimar-type fancoil command to manage 2-pipe fan coils. | | | | |
| **DPTx_Fancoil4Pipes** | **DP_ControlFancoil_4T** | DPT_Vimar_Fancoil_4T | 2 | W |
| Receipt of Vimar-type fancoil command to manage 4-pipe fan coils. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.7 Functional block FBT_DIMMER_W
Name: FBT_DIMMER_W

#### Functional description

This functional block controls the lights. You can change the control by configuring its parameters.



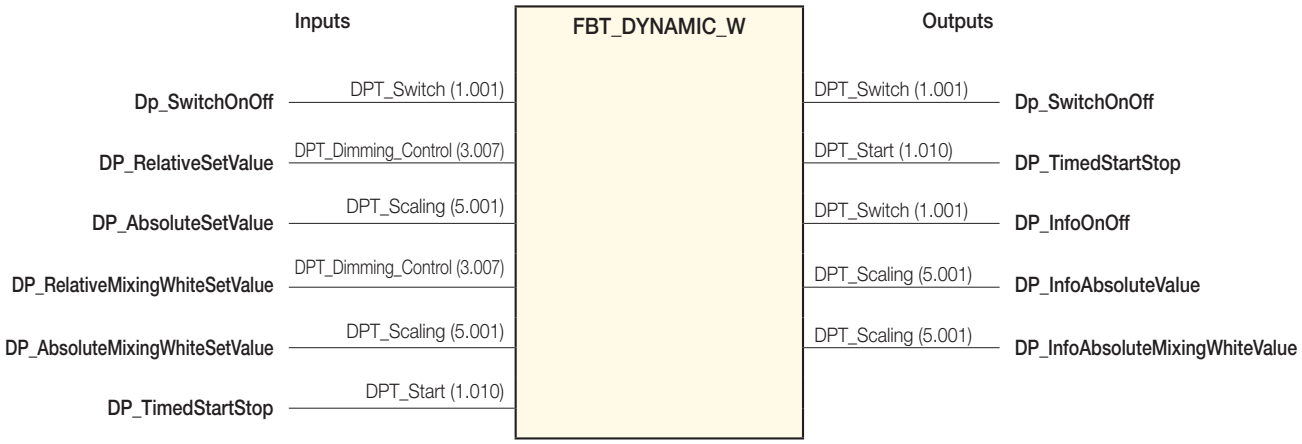

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **Dp_SwitchOnOff** | DPT_Switch (1.001) | 1 | RWT |
| Dimmer on-off control with fading effect (1=On,0=OFF). | | | | |
| **DPTx_DimmingControl** | **DP_RelativeSetValue** | DPT_Dimming_Control (3.007) | 2 | W |
| Message received as a request to increase or decrease the dimmer output level. | | | | |
| **DPTx_Brightness** | **DP_AbsoluteSetValue** | DPT_Scaling (5.001) | 3 | W |
| Message containing the absolute control value to move the output to. | | | | |
| **DPTx_TimedStartStop** | **DP_TimedStartStop** | DPT_Start (1.010) | 4 | RWT |
| Timer-operated one- position stable dimmer on-off control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 6 | RT |
| Current value of the output status. | | | | |
| **DPTx_BrightnessInfo** | **DP_InfoAbsoluteValue** | DPT_Scaling (5.001) | 7 | RT |
| Current value of the output absolute. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.8 Functional block FBT_DYNAMIC_W
Name: FBT_DYNAMIC_W

#### Functional description

This functional block controls the lights. You can change the control by configuring its parameters.



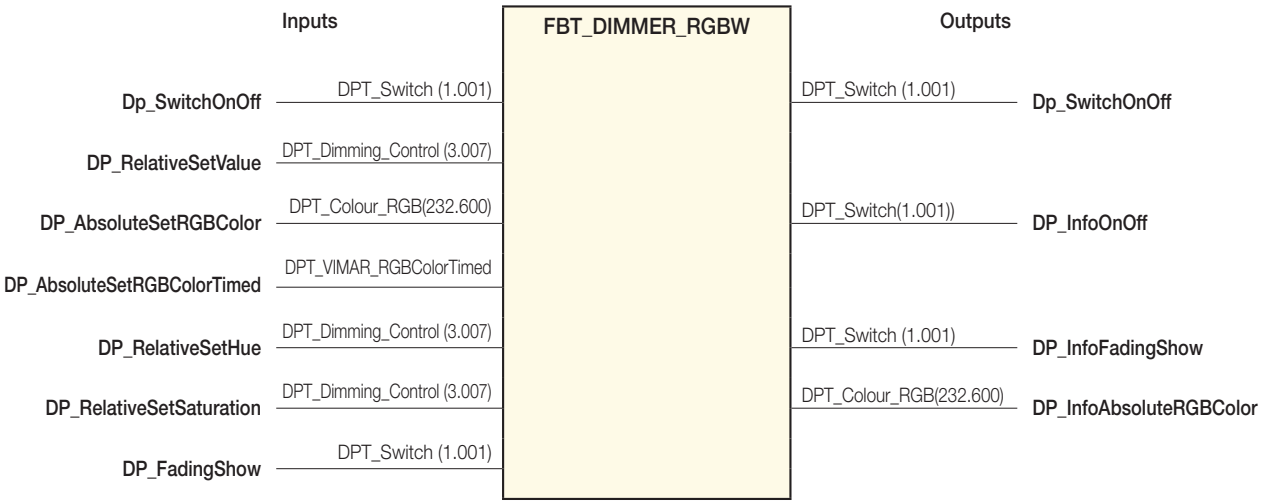

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **Dp_SwitchOnOff** | DPT_Switch (1.001) | 1 | RWT |
| Dimmer on-off control with fading effect (1=On,0=OFF). | | | | |
| **DPTx_DimmingControl** | **DP_RelativeSetValue** | DPT_Dimming_Control (3.007) | 2 | W |
| Message received as a request to increase or decrease the dimmer output level. | | | | |
| **DPTx_Brightness** | **DP_AbsoluteSetValue** | DPT_Scaling (5.001) | 3 | W |
| Message containing the absolute control value to move the output to. | | | | |
| **DPTx_RelativeMixingWhiteValue** | **DP_RelativeMixingWhiteSetValue** | DPT_Dimming_Control (3.007) | 4 | W |
| Message received as a request to increase or reduce the mixing value between warm white and cold white levels in the output channels. | | | | |
| **DPTx_AbsoluteMixingWhiteValue** | **DP_AbsoluteMixingWhiteSetValue** | DPT_Scaling (5.001) | 5 | W |
| Message containing the mixing value between warm white and cold white levels the relative channels must move to.<br>0%: warm white<br>100%: cold white | | | | |
| **DPTx_TimedStartStop** | **DP_TimedStartStop** | DPT_Start (1.010) | 6 | RWT |
| Timer-operated one- position stable dimmer on-off control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 8 | RT |
| Current value of the output status. | | | | |
| **DPTx_BrightnessInfo** | **DP_InfoAbsoluteValue** | DPT_Scaling (5.001) | 9 | RT |
| Current value of the output absolute. | | | | |
| **DPTx_InfoAbsoluteMixingWhiteValue** | **DP_InfoAbsoluteMixingWhiteValue** | DPT_Scaling (5.001) | 10 | RT |
| Mixing value between warm white and cold white levels. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.9 Functional block FBT_DIMMER_RGBW
Name: FBT_DIMMER_RGB

#### Functional description

This functional block controls the lights. You can change the control by configuring its parameters.




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **Dp_SwitchOnOff** | DPT_Switch (1.001) | 1 | RWT |
| Dimmer on-off control with fading effect (1=On,0=OFF). | | | | |
| **DPTx_DimmingControl** | **DP_RelativeSetValue** | DPT_Dimming_Control (3.007) | 2 | W |
| Message received as a request to increase or decrease the dimmer output level. | | | | |
| **DPTx_RgbColour** | **DP_AbsoluteSetRGBColor** | DPT_Colour_RGB(232.600) | 3 | W |
| Absolute value of the RGB coordinate to be set on the actuator. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch(1.001) | 5 | RT |
| Current value of the dimmer status. | | | | |
| **DPTx_TimedRgbColour** | **DP_AbsoluteSetRGBColorTimed** | DPT_VIMAR_RGBColorTimed | 6 | W |
| Custom message similar to DP_AbsoluteSetRGBColor containing the absolute value of the RGB coordinates the dimmer has to move to, plus the possibility to define the time taken to actuate the required value. | | | | |
| **DPTx_ControlHue** | **DP_RelativeSetHue** | DPT_Dimming_Control (3.007) | 7 | W |
| The datapoint starts the control of the HUE value, i.e. the colour to be reproduced. | | | | |
| **DPTx_ControlSaturation** | **DP_RelativeSetSaturation** | DPT_Dimming_Control (3.007) | 8 | W |
| The datapoint starts the saturation control. | | | | |
| **DPTx_FadingShow** | **DP_FadingShow** | DPT_Switch (1.001) | 9 | W |
| Activation of the FadingShow procedure. | | | | |
| **DPTx_FadingShowInfo** | **DP_InfoFadingShow** | DPT_Switch (1.001) | 10 | RT |
| Current value of the FadingShow status sent on change. | | | | |
| **DPTx_InfoAbsoluteRGBColor** | **DP_InfoAbsoluteRGBColor** | DPT_Colour_RGB(232.600) | 11 | RT |
| Value of the RGB coordinates set in the actuator. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.10 Functional block FBT_ACT_DIMMER2_RGB_W
Name: FBT_ACT_DIMMER2_RGB_W

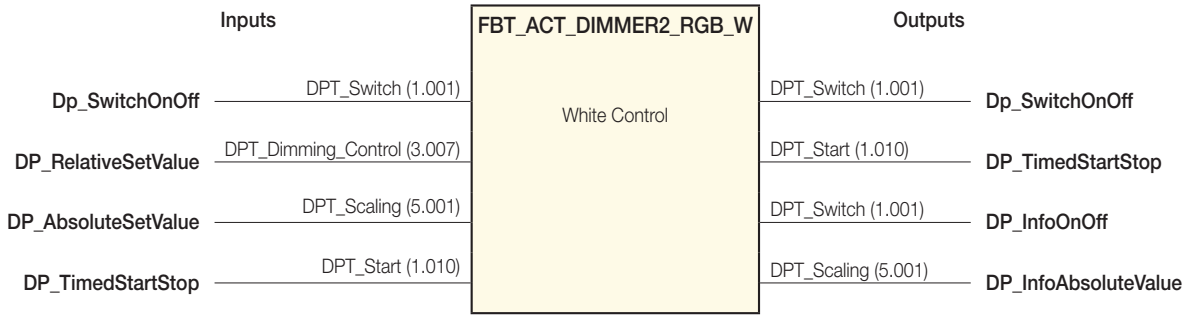
#### Functional description

This functional block controls the lights. You can change the control by configuring its parameters.

#### N.B.

FBT_ACT_DIMMER2_RGB_W presents 2 different group object configurations depending on how it is configured in VIEW Pro. Here below the two different modes are described: White Control and Colour Control.

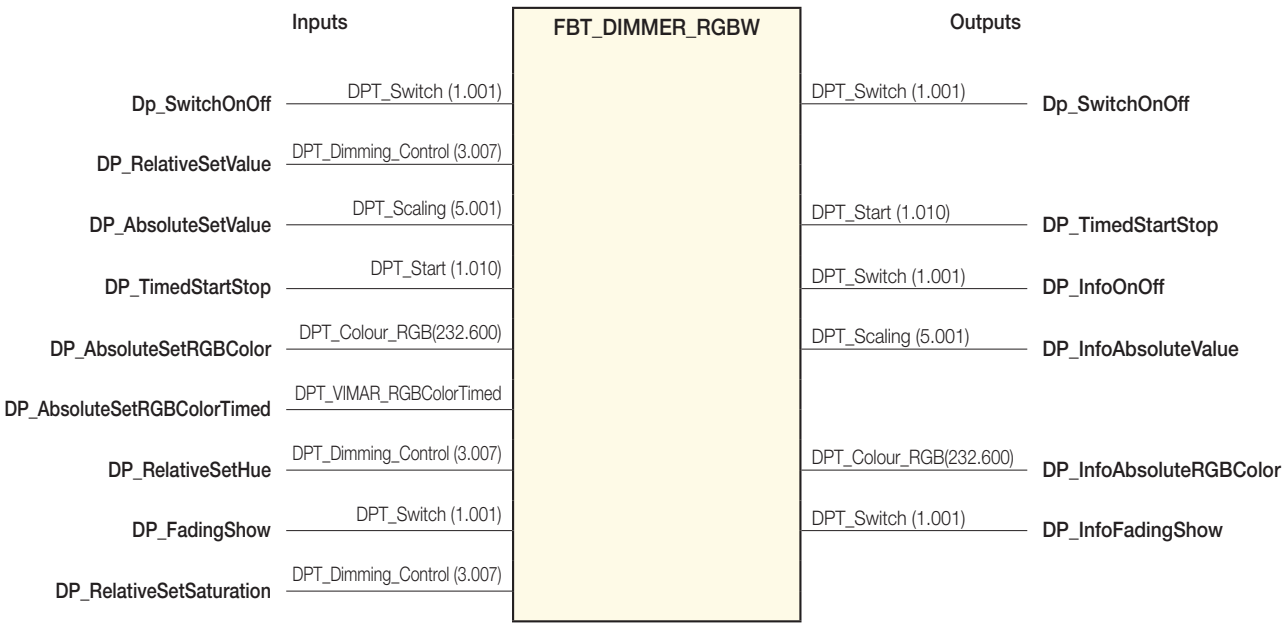
#### 4.10.1 Functional block FBT_ACT_DIMMER2_RGB_W: White Control




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **Dp_SwitchOnOff** | DPT_Switch (1.001) | 1 | RWT |
| Dimmer on-off control with fading effect (1=On,0=OFF). | | | | |
| **DPTx_DimmingControl** | **DP_RelativeSetValue** | DPT_Dimming_Control (3.007) | 2 | W |
| Message received as a request to increase or decrease the dimmer output level. | | | | |
| **DPTx_Brightness** | **DP_AbsoluteSetValue** | DPT_Scaling (5.001) | 3 | W |
| Message containing the absolute control value to move the output to. | | | | |
| **DPTx_TimedStartStop** | **DP_TimedStartStop** | DPT_Start (1.010) | 4 | RWT |
| Timer-operated one- position stable dimmer on-off control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 6 | RT |
| Current value of the output status. | | | | |
| **DPTx_BrightnessInfo** | **DP_InfoAbsoluteValue** | DPT_Scaling (5.001) | 7 | RT |
| Current value of the output absolute. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

### 4.10.2 Functional block FBT_ACT_DIMMER2_RGB_W: Colour Control




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **Dp_SwitchOnOff** | DPT_Switch (1.001) | 1 | RWT |
| Dimmer on-off control with fading effect (1=On,0=OFF). | | | | |
| **DPTx_DimmingControl** | **DP_RelativeSetValue** | DPT_Dimming_Control (3.007) | 2 | W |
| Message received as a request to increase or decrease the dimmer output level. | | | | |
| **DPTx_Brightness** | **DP_AbsoluteSetValue** | DPT_Scaling (5.001) | 3 | W |
| Message containing the absolute control value to move the output to. | | | | |
| **DPTx_TimedStartStop** | **DP_TimedStartStop** | DPT_Start (1.010) | 4 | RWT |
| Timer-operated one- position stable dimmer on-off control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 6 | RT |
| Current value of the output status. | | | | |
| **DPTx_BrightnessInfo** | **DP_InfoAbsoluteValue** | DPT_Scaling (5.001) | 7 | RT |
| Current value of the output absolute. | | | | |
| **DPTx_RgbColour** | **DP_AbsoluteSetRGBColor** | DPT_Colour_RGB(232.600) | 8 | W |
| Absolute value of the RGB coordinate to be set on the actuator. | | | | |
| **DPTx_TimedRgbColour** | **DP_AbsoluteSetRGBColorTimed** | DPT_VIMAR_RGBColorTimed | 9 | W |
| Custom message similar to DP_AbsoluteSetRGBColor containing the absolute value of the RGB coordinates the dimmer has to move to, plus the possibility to define the time taken to actuate the required value. | | | | |
| **DPTx_ControlHue** | **DP_RelativeSetHue** | DPT_Dimming_Control (3.007) | 10 | W |
| The datapoint starts the control of the HUE value, i.e. the colour to be reproduced. | | | | |
| **DPTx_FadingShow** | **DP_FadingShow** | DPT_Switch (1.001) | 11 | W |
| Activation of the FadingShow procedure. | | | | |

## SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ControlSaturation** | **DP_RelativeSetSaturation** | DPT_Dimming_Control (3.007) | 12 | W |
| The datapoint starts the saturation control. | | | | |
| **DPTx_InfoAbsoluteRGBColor** | **DP_InfoAbsoluteRGBColor** | DPT_Colour_RGB(232.600) | 13 | RT |
| Value of the RGB coordinates set in the actuator. | | | | |
| **DPTx_FadingShowInfo** | **DP_InfoFadingShow** | DPT_Switch (1.001) | 14 | RT |
| Current value of the FadingShow status sent on change. | | | | |

# Index

1 Introduction ..... 58
2 Overview of the By-me energy management system ..... 58
3 Devices 59
3.1 Load control module (01455) ..... 59
3.1.1 FB_CCDeviceManagement ..... 59
3.1.1.1 Functional Block ..... 59
3.1.1.2 Description of Communication Objects ..... 59
3.1.2 FB_LineManagement ..... 60
3.1.2.1 Channel 0 ..... 60
3.1.2.2 Channel 1 ..... 60
3.1.2.3 Description of Communication Objects ..... 60
3.1.3 FB_Control ..... 61
3.1.3.1 Channel 1 ..... 61
3.1.3.2 Description of Communication Objects ..... 61
3.1.4 FB_CCPowerMeter ..... 62
3.1.4.1 Channel 0 ..... 62
3.1.4.2 Description of Communication Objects ..... 62
3.2 3-input 1M/DIN power meter (01450) – 1M flush mounting power meter (01451) ..... 63
3.2.1 FB_PowerMeter ..... 63
3.2.1.1 Functional Block ..... 63
3.2.1.2 Description of Communication Objects ..... 63
3.3 1M/DIN relay with current sensor (01456) ..... 64
3.3.1 FB_ActuatorWithDiagnostics ..... 64
3.3.1.1 Functional Block ..... 64
3.3.1.2 Description of Communication Objects ..... 65
3.3.2 FB_PowerMeter ..... 65
3.4 2M Relay with flush mounting current sensor (14537, 19537, 20537) ..... 65
3.4.1 FB_Actuator ..... 65
3.4.1.1 Functional Block ..... 65
3.4.1.2 Description of Communication Objects ..... 66
3.4.2 FB_PowerMeter ..... 66
3.5 Pulse counter (01452) ..... 66
3.5.1 FB_Counter ..... 66
3.5.1.1 Functional Block ..... 66
3.4.1.2 Description of Communication Objects ..... 67

# SECTION 3: ENERGY MANAGEMENT

## 1. Introduction

The document describes the features of the KNX communication interfaces of the By-me energy management system devices, the method for obtaining the system configuration information and application examples for controlling and commanding system equipment via standard KNX messages.

This document does not describe the functional, technical and regulatory characteristics of the energy management system, for which you should refer to the official documentation

## 2. Overview of the By-me energy management system

The By-me devices involved in an energy management system are:
- Load control module (01455);
- 3-input power meter (01450);
- 1M relay with current sensor on DIN rail (01456);
- 1M flush mounting power meter (01451);
- 2M flush mounting relay with current sensor (14537, 19537, 20537);
- Pulse counter (01452).

Refer to the By-me Plus system manual – Energy management section.

The following table shows the map of generated/drawn power information according to the various possible system configurations.

| | |
|---|---|
| **Single-phase system with no generation**<br>Refer to the By-me Plus Installation manual for the diagram. | Load control module 01455<br>BLOCK By-me Power Meter 1: Node P_ACT = Power Exchanged<br>Note: > 0 = Drawn off; < 0 Fed in<br>The By-me block Line 1 is not used for metering. |
| **Single-phase system with "local" generation**<br>Refer to the By-me Plus Installation manual for the diagram. | Load control module 01455<br>BLOCK By-me Power Meter 1: Node P_ACT = Power Exchanged<br>Note: > 0 = Drawn off; < 0 Fed in<br>BLOCK By-me Power Meter 2: Node P_ACT = Power Generated<br>Note: must be ≥ 0<br>The By-me block Line 1 and Line 2 are not used for metering. |
| **Single-phase system with "remote" generation**<br>Refer to the By-me Plus Installation manual for the diagram. | Load control module 01455<br>BLOCK By-me Line 1: Node P_ACT = Power Generated<br>Note: must be ≥ 0<br>BLOCK By-me Power Meter 1: Node P_ACT = Power Exchanged<br>Note: > 0 = Drawn off; < 0 Fed in<br>Power meter 01450<br>BLOCK By-me Power Meter 1: Node P_ACT = Power Generated<br>Note: must be ≥ 0 |
| **Three-phase system without production**<br>Refer to the By-me Plus Installation manual for the diagram. | Load control module 01455<br>BLOCK By-me Power Meter 1: Node P_ACT = Power Exchanged<br>Note: > 0 = Drawn off; < 0 Fed in<br>The By-me block Line 1 is not used for metering.<br>BLOCK By-me Power Meter 2: Node P_ACT = Power Exchanged<br>Note: > 0 = Drawn off; < 0 Fed in<br>The By-me block Line 2 is not used for metering.<br>BLOCK By-me Power Meter 3: Node P_ACT = Power Exchanged<br>Note: > 0 = Drawn off; < 0 Fed in<br>The By-me block Line 3 is not used for metering. |
| **Three-phase system with generation (from one to three phases)**<br>Refer to the By-me Plus Installation manual for the diagram. | Load control module 01455<br>BLOCK By-me Line 1-2-3: Node P_ACT = Power Generated<br>Note: must be ≥ 0<br>BLOCK By-me Meter 1-2-3: Node P_ACT = Power Exchanged<br>Note: > 0 = Drawn off; < 0 Fed in<br>Power meter 01450<br>BLOCK By-me Meter 1-2-3: Node P_ACT = Power Generated<br>Note: must be ≥ 0 |

# SECTION 3: ENERGY MANAGEMENT

## 3. Devices

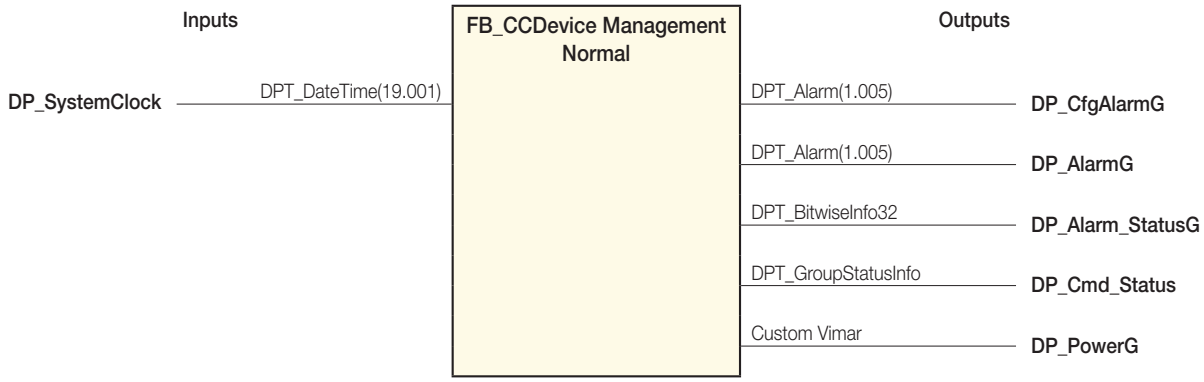
### 3.1 Load control module (01455)

The device consists of 23 Functional Blocks:

| Fb Index | Fb Name | Fb Type |
|---|---|---|
| 1 | CCDeviceManagement) | FB_CCDeviceManagement (0x4C) |
| The FB provides datapoints and parameters for the entire device. | | |
| 2 - 4 | LineManagement | FB_LineManagement (0x4E) |
| The FB implements the logic to decode the data from meters connected to the bus providing the logic device with the information. Paired to lines L1, L2 and L3. | | |
| 5 - 20 | Controls | FB_Control (0x4F) |
| The FB decides how to control the paired load according to the request received from the application logic conditioning it with the forced states. | | |
| 21 - 23 | Power Meter on CC | FB_CCPowerMeter (0x4D) |
| The FB implements the power meter of the TA connected to the device and provides information on the power output, energy, current and voltage it measures. Paired to lines L1, L2 and L3 | | |

#### 3.1.1 FB_CCDeviceManagement

##### 3.1.1.1 Functional Block




##### 3.1.1.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

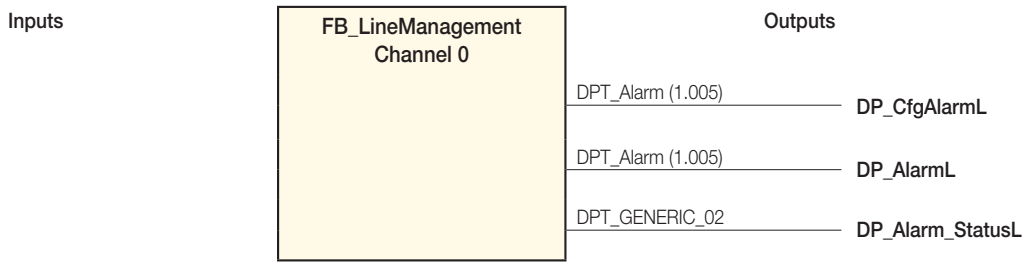
| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| DPTx_CfgGlobalAlarm | DP_CfgAlarmG | DPT_Alarm (DPT_ID = 1.005) | 1 | RT |
| Signals the presence of an alarm condition according to the alarms selected with PAR_AlarmMask | | | | |
| DPTx_GlobalAlarm | DP_AlarmG | DPT_Alarm (DPT_ID = 1.005) | 2 | RT |
| Signals the presence of an alarm condition for the whole load control system. | | | | |
| DPTx_GlobalAlarmStatus | DP_Alarm_StatusG | DPT_BitwiseInfo32 | 3 | RT |
| Gives information on the alarm status. | | | | |
| DPTx_CmdStatus | DP_Cmd_Status | DPT_GroupStatusInfo | 4 | RT |
| Identifies the resulting status of the 16 control groups. | | | | |
| DPTx_SysClockDateTime | DP_SystemClock | DPT_DateTime(19.001) | 5 | WT |
| Updates the device clock.<br>Used for the time schedule.<br>When the device is started it asks the system for the time, if this does not arrive it works with its own internal clock | | | | |
| DPTx_GlobalPower | DP_PowerG | Custom Vimar | 6 | R |
| Gives the values of the power active on the three network lines. | | | | |

## SECTION 3: ENERGY MANAGEMENT

### 3.1.2 FB_LineManagement

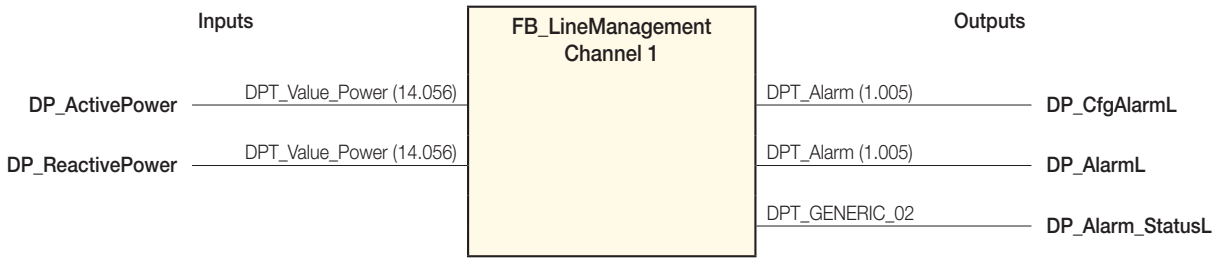
#### 3.1.2.1 Channel 0

##### 3.1.2.1.1 Functional Block




#### 3.1.2.2 Channel 1

##### 3.1.2.2.1 Functional Block




#### 3.1.2.3 Description of Communication Objects

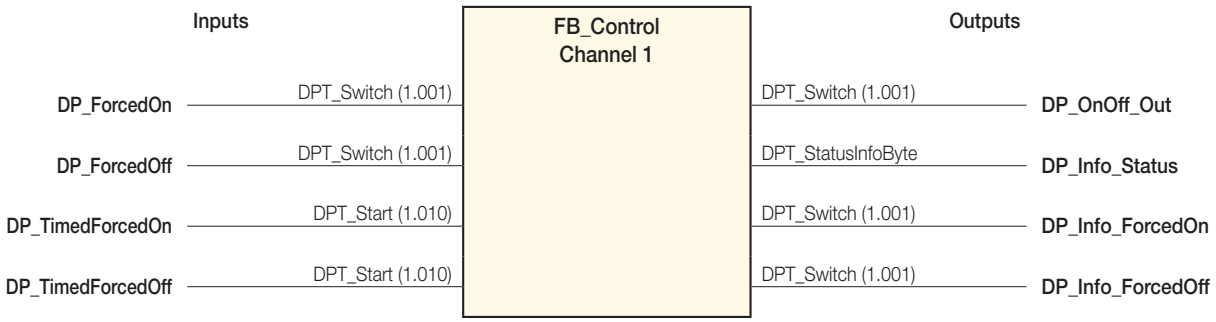
| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ActivePower** | **DP_ActivePower** | DPT_Value_Power(DPT_ID = 14.056) | 1 | W |
| The group object is linked to the datapoint DP_PowerA by a remote power meter in the system. The power value has a resolution of 1W. | | | | |
| **DPTx_ReactivePower** | **DP_ReactivePower** | DPT_Value_Power(DPT_ID = 14.056) | 2 | W |
| The group object is linked to the datapoint DP_PowerR by a remote power meter in the system. In the case of three-phase installations with production, the DP is used for the measurement of the reactive energy generated on a phase. The power value has a resolution of 1W. | | | | |
| **DPTx_LineCfgAlarm** | **DP_CfgAlarmL** | DPT_Alarm (DPT_ID = 1.005) | 3 | RT |
| Signals the presence of an alarm condition according to the alarms selected with PAR_AlarmMask | | | | |
| **DPTx_LineAlarm** | **DP_AlarmL** | DPT_Alarm (DPT_ID = 1.005) | 4 | RT |
| Signals the presence of an alarm condition for the whole load control system. | | | | |
| **DPTx_LineAlarmStatus** | **DP_Alarm_StatusL** | DPT_GENERIC_02 | 5 | RT |
| Gives information on the status of the alarms configured via PAR_AlarmMask (device parameter) | | | | |

# SECTION 3: ENERGY MANAGEMENT

### 3.1.3 FB_Control

#### 3.1.3.1 Channel 1

##### 3.1.3.1.1 Functional Block




#### 3.1.3.2 Description of Communication Objects

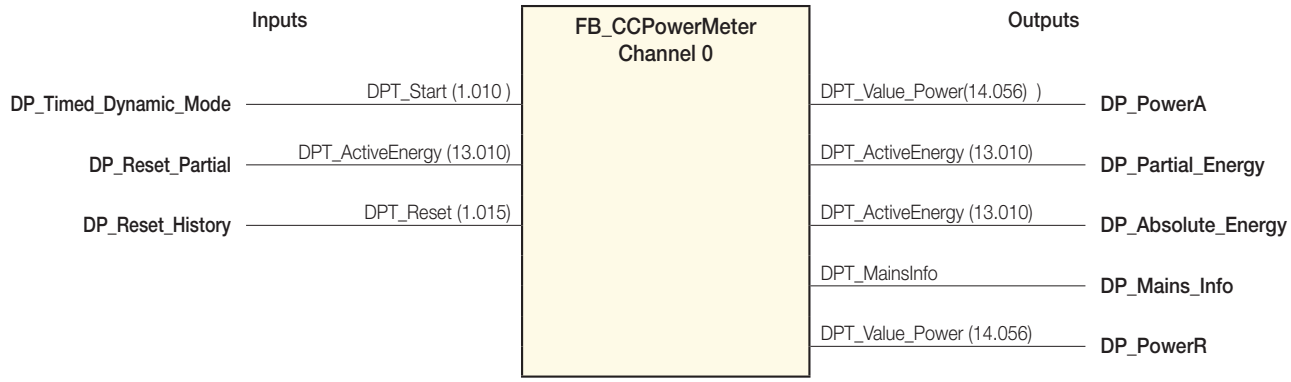
| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_LoadManagerForcedOn** | **DP_ForcedOn** | DPT_Switch (DPT_ID = 1.001) | 1 | WT |
| The message is received at the input to force the paired relay On. Upon receiving Off the control returns to the load control module. It is not KNX standard as KNX would have 2 bits of data while our rocker buttons have only one bit. NB: upon receipt of a ForcedOn any ForcedOff must return to Off. | | | | |
| **DPTx_LoadManagerForcedOff2** | **DP_ForcedOff** | DPT_Switch (DPT_ID = 1.001) | 2 | WT |
| The message is received at the input to force the paired relay Off. Upon receiving Off the control returns to the load control module. It is not KNX standard as KNX would have 2 bits of data while our rocker buttons have only one bit. NB: Upon receipt of a ForcedOff any ForcedOn must return to Off. | | | | |
| **DPTx_LoadManagerTimedForcedOn** | **DP_TimeForcedOn** | DPT_Start (DPT_ID = 1.010) | 3 | WT |
| The message is received at the input to force the paired relay On for a time determined by the parameter "Time Forced On." | | | | |
| **DPTx_LoadManagerTimedForcedOff** | **DP_TimeForcedOff** | DPT_Start (DPT_ID = 1.010) | 4 | WT |
| The message is received at the input to force the paired relay Off for a time determined by the parameter "Time Forced Off." | | | | |
| **DPTx_LoadManagerOnOff** | **DP_OnOff_Out** | DPT_Switch (DPT_ID = 1.001) | 5 | T |
| Controls activating/deactivating the paired actuator | | | | |
| **DPTx_StatusInfo** | **DP_Info_Status** | (DPT_StatusInfoByte) | 6 | RT |
| Enclosure for measurement data: Voltage, Current, Frequency, Power Factor aggregated in a merged Datapoint. The message is sent according to the policy implemented by PAR_TxOthers | | | | |
| **DPTx_ForcedOnInfo** | **DP_info_forcedon** | DPT_Value_Power (DPT_ID = 14.056) | 7 | RT |
| The message is sent as a result of a substantial change (determined by the PowerR transmission threshold) in the Reactive power on the line paired with the FB. In addition, the reactive power can also be transmitted in time (parameter "PowerR transmission timer"). The power value has a resolution of 1 var. | | | | |
| **DPTx_ForcedOffInfo** | **DP_info_forcedoff** | DPT_Start (DPT_ID = 1.010 ) | 8 | W |
| Starts or stops the Timed Dynamic Mode for greater power data visualization accuracy. | | | | |

# SECTION 3: ENERGY MANAGEMENT

### 3.1.4 FB_CCPowerMeter

#### 3.1.4.1 Channel 0

##### 3.1.4.1.1 Functional Block




#### 3.1.4.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_PartialResetEnergy** | **DP_Reset_Partial** | DPT_ActiveEnergy (DPT_ID = 13.010) | 1 | W |
| The message is used to reset the partial counter of the energy consumed, initialising it to the value contained in the message. | | | | |
| **DPTx_HistoryReset** | **DP_Reset_History** | DPT_Reset (DPT_ID = 1.015) | 2 | W |
| The message is used to reset the entire log stored in the device, partial energy and absolute energy. | | | | |
| **DPTx_ActivePower** | **DP_PowerA** | DPT_Value_Power(DPT_ID = 14.056) | 3 | RT |
| The message is sent as a result of a substantial change (determined by the power transmission threshold) in the power on the line paired with the FB. In addition, the power can also be transmitted in time (parameter "Power send timer"). The power value has a resolution of 1W. | | | | |
| **DPTx_PartialEnergy** | **DP_Partial_Energy** | DPT_ActiveEnergy (DPT_ID = 13.010) | 4 | RT |
| The message is sent as a result of a substantial change (determined by the energy transmission threshold) in the energy (partially totalled) on the line paired with the FB. In addition, the energy can also be transmitted in time (parameter "Energy transmission timer"). The energy value has a resolution of 1Wh. This is a partial counter, reset by a writing on the DP_Reset_Partial. | | | | |
| **DPTx_AbsoluteEnergy** | **DP_Absolute_Energy** | DPT_ActiveEnergy (DPT_ID = 13.010) | 5 | RT |
| The message is sent as a result of a substantial change (determined by the energy transmission threshold) in the energy (absolute totalled) on the line paired with the FB. In addition, the energy can also be transmitted in time (parameter "Energy transmission timer"). The energy value has a resolution of 1Wh. The energy value has a resolution of 1Wh. This is a counter reset by a writing on the DP_Reset_History. | | | | |
| **DPTx_MainsInfo** | **DP_Mains_Info** | DPT_MainsInfo | 6 | RT |
| Enclosure for measurement data: Voltage, Current, Frequency, Power Factor aggregated in a merged Datapoint. The message is sent according to the policy implemented by PAR_TxOthers | | | | |
| **DPTx_ReactivePower** | **DP_PowerR** | DPT_Value_Power (DPT_ID = 14.056) | 7 | RT |
| The message is sent as a result of a substantial change (determined by the PowerR transmission threshold) in the Reactive power on the line paired with the FB. In addition, the reactive power can also be transmitted in time (parameter "PowerR transmission timer"). The power value has a resolution of 1 var. | | | | |
| **DPTx_TimedDynamicMode** | **DP_Timed_Dynamic_Mode** | DPT_Start (DPT_ID = 1.010 ) | 8 | W |
| Starts or stops the Timed Dynamic Mode for greater power data visualization accuracy. | | | | |

# SECTION 3: ENERGY MANAGEMENT

### 3.2 3-input 1M/DIN power meter (01450) – 1M flush mounting power meter (01451)

The devices contain the following functional blocks:

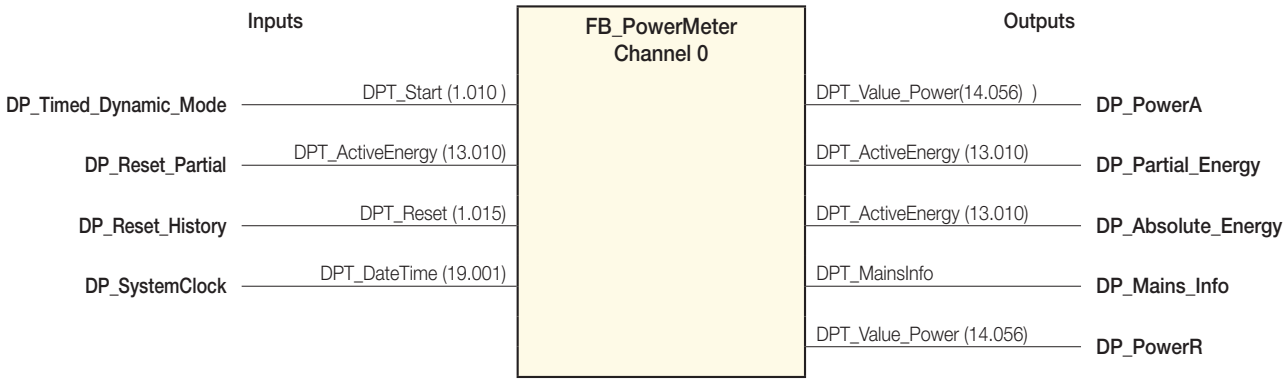
| Description | FB Type ID |
|---|---|
| Power Meter | 0x51 |
| The FB implements the power meter implemented in the device and provides information on the power output, energy, current and voltage it measures. | |

In the case of the 3-input 1M/DIN power meter (01450) there are three instances of the functional block.

#### 3.2.1 FB_PowerMeter

##### 3.2.1.1 Functional Block

###### 3.2.1.1.1 Channel 0




##### 3.2.1.2 Description of Communication Objects

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_PartialResetEnergy** | **DP_Reset_Partial** | DPT_ActiveEnergy (DPT_ID = 13.010) | 1 | W |
| The message is used to reset the partial counter of the energy consumed, initialising it to the value contained in the message. | | | | |
| **DPTx_HistoryReset** | **DP_Reset_History** | DPT_Reset (DPT_ID = 1.015) | 2 | W |
| The message is used to reset the entire log stored in the device, partial energy and absolute energy. | | | | |
| **DPTx_ActivePower** | **DP_PowerA** | DPT_Value_Power(DPT_ID = 14.056) | 3 | RT |
| The message is sent as a result of a substantial change (determined by the power transmission threshold) in the power on the line paired with the FB. In addition, the power can also be transmitted in time (parameter "Power send timer"). The power value has a resolution of 1W. | | | | |
| **DPTx_PartialEnergy** | **DP_Partial_Energy** | DPT_ActiveEnergy (DPT_ID = 13.010) | 4 | RT |
| The message is sent as a result of a substantial change (determined by the energy transmission threshold) in the energy (partially totalled) on the line paired with the FB. In addition, the energy can also be transmitted in time (parameter "Energy transmission timer"). The energy value has a resolution of 1Wh. This is a partial counter, reset by a writing on the DP_Reset_Partial. | | | | |
| **DPTx_AbsoluteEnergy** | **DP_Absolute_Energy** | DPT_ActiveEnergy (DPT_ID = 13.010) | 5 | RT |
| The message is sent as a result of a substantial change (determined by the energy transmission threshold) in the energy (absolute totalled) on the line paired with the FB. In addition, the energy can also be transmitted in time (parameter "Energy transmission timer"). The energy value has a resolution of 1Wh. The energy value has a resolution of 1Wh. This is a counter reset by a writing on the DP_Reset_History. | | | | |
| **DPTx_MainsInfoWithoutEnergy** | **DP_Mains_Info** | DPT_MainsInfo | 6 | RT |
| Enclosure for measurement data: Voltage, Current, Frequency, Power Factor aggregated in a merged Datapoint. The message is sent according to the policy implemented by PAR_TxOthers | | | | |

## SECTION 3: ENERGY MANAGEMENT

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ReactivePower** | **DP_PowerR** | DPT_Value_Power (DPT_ID = 14.056) | 7 | RT |
| The message is sent as a result of a substantial change (determined by the PowerR transmission threshold) in the Reactive power on the line paired with the FB. In addition, the reactive power can also be transmitted in time (parameter "PowerR transmission timer"). The power value has a resolution of 1 var. | | | | |
| **DPTx_TimedDynamicMode** | **DP_Timed_Dynamic_Mode** | DPT_Start (DPT_ID = 1.010 ) | 8 | W |
| Starts or stops the Timed Dynamic Mode for greater power data visualization accuracy. | | | | |
| **DPTx_SysClockDateTime** | **DP_SystemClock** | DPT_DateTime(19.001) | 12 | WT |
| Updates the device clock.<br>Used for the time schedule.<br>When the device is started it asks the system for the time, if this does not arrive it works with its own internal clock. | | | | |

### 3.3 1M/DIN relay with current sensor (01456)

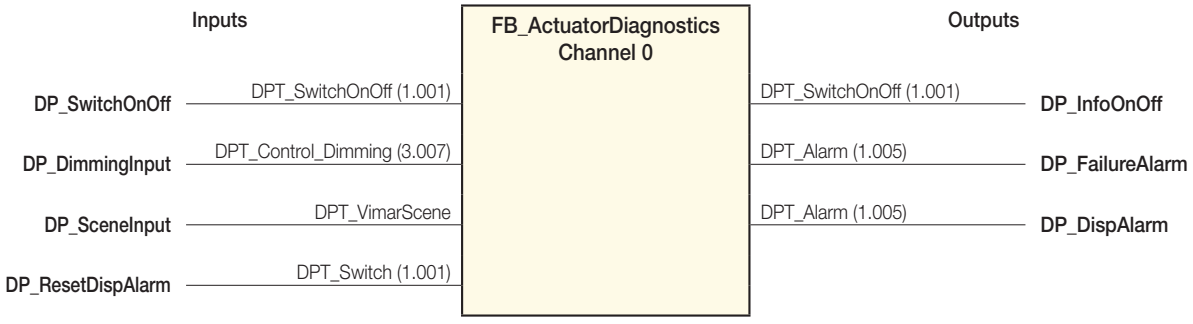
The device contains the following functional blocks:

| FB Index | FB Type | FB Type ID |
|---|---|---|
| 1 | FB Actuator with diagnostics | 0x53 |
| The FB implements load actuation and signals any alarms | | |
| 2 | FB Power Meter | 0x51 |
| The FB implements the power meter via shunts connected to the device and provides information on the power output, energy, current and voltage it measures. | | |

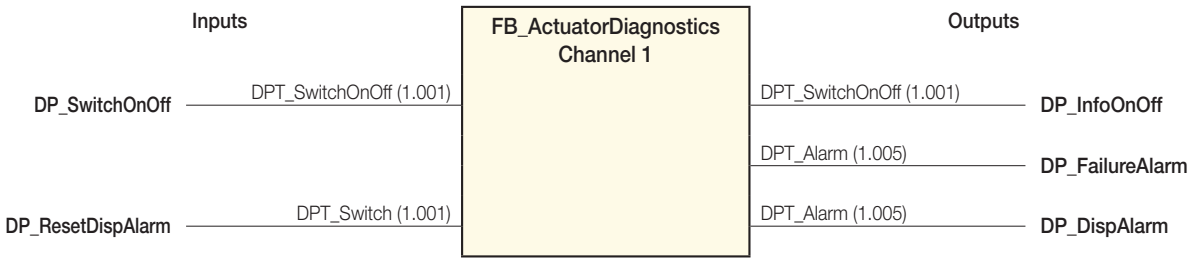
#### 3.3.1 FB_ActuatorWithDiagnostics

##### 3.3.1.1 Functional Block

###### 3.3.1.1.1 Channel 0




###### 3.3.1.1.2 Channel 1



## SECTION 3: ENERGY MANAGEMENT

#### 3.3.1.2 Description of Communication Objects

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_SwitchOnOff** | DPT_Switch (DPT_ID = 1.001) | 1 | WT |
| Actuator on and off control. | | | | |
| **DPTx_DimmingControl** | **DP_DimmingInput** | DPT_Control_Dimming(DPT_ID = 3.007) | 2 | W |
| Control with management of the actuator operating modes. | | | | |
| **DPTx_DispersionAlarmReset** | **DP_ResetDispAlarm** | DPT_Switch (DPT_ID = 1.001) | 4 | W |
| Dispersion alarm reset control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (DPT_ID = 1.001) | 5 | RT |
| Datapoint informing of the change of actuator status. | | | | |
| **DPTx_FailureAlarm** | **DP_FailureAlarm** | DPT_Alarm (DPT_ID = 1.005) | 6 | RT |
| Datapoint informing of the alarm status (no current absorption when the actuator is on. | | | | |
| **DPTx_DispersionAlarm** | **DP_DispAlarm** | DPT_Alarm (DPT_ID = 1.005) | 7 | RT |
| Current dispersion alarm on the load | | | | |

#### 3.3.2 FB_PowerMeter

v. 3.2.1

### 3.4 Relay with flush mounting 2M current sensor (14537, 19537, 20537)

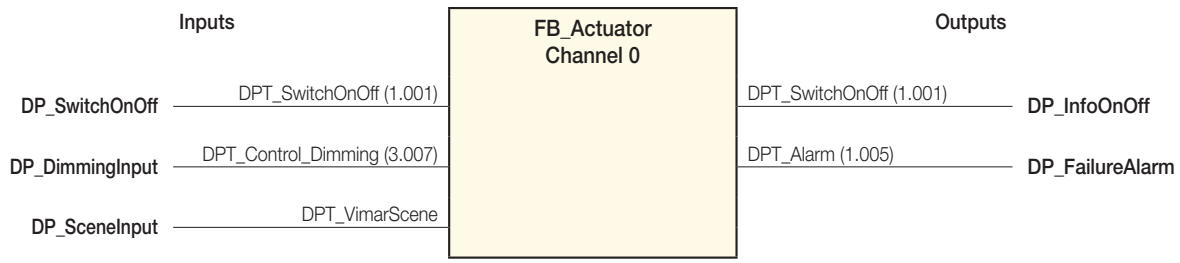
The device contains the following functional blocks:

| FB Index | FB Type | FB Type ID |
|---|---|---|
| 1 | FB Actuator | 0x54 |
| The FB implements the load actuation | | |
| 2 | FB Power Meter | 0x51 |
| The FB implements the power meter via shunts connected to the device and provides information on the power output, energy, current and voltage it measures. | | |

#### 3.4.1 FB_Actuator

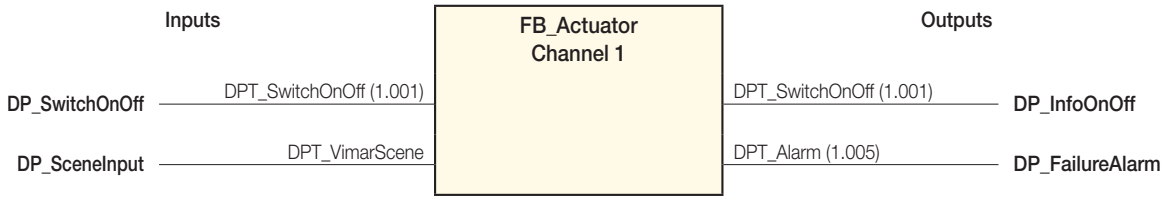
##### 3.4.1.1 Functional Block

###### 3.4.1.1.1 Channel 0



## SECTION 3: ENERGY MANAGEMENT

3.4.1.1.2 Channel 1




3.4.1.2 Description of Communication Objects

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_SwitchOnOff** | DPT_Switch (DPT_ID = 1.001) | 1 | W |
| Actuator on and off control. | | | | |
| **DPTx_DimmingControl** | **DP_DimmingInput** | DPT_Control_Dimming(DPT_ID = 3.007) | 2 | W |
| Control with management of the actuator operating modes. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (DPT_ID = 1.001) | 4 | RT |
| Datapoint informing of the change of actuator status. | | | | |
| **DPTx_DispersionAlarm** | **DP_FailureAlarm** | DPT_Alarm (DPT_ID = 1.005) | 5 | RT |
| Datapoint informing of the alarm status (no current absorption when the actuator is on. | | | | |

3.4.2 FB_PowerMeter

v. 3.2.1

3.5 Pulse counter (01452)

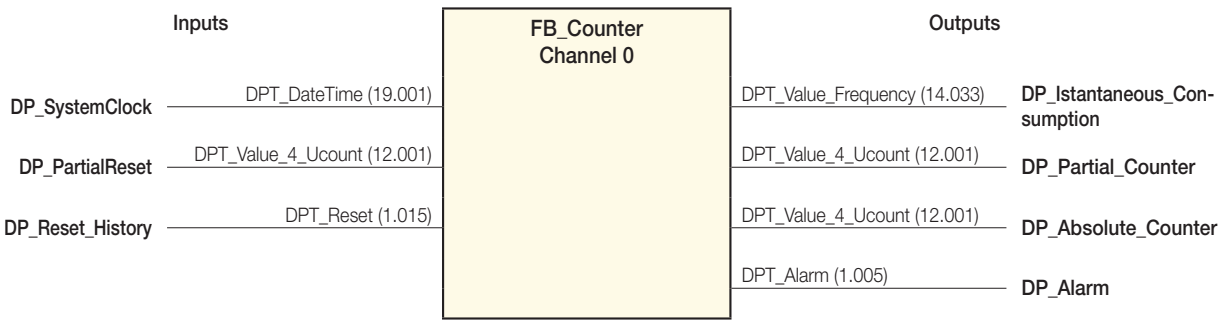
The device contains a functional block

| FB Index | Description | FB Type ID |
|---|---|---|
| 1 | Counter | 0x55 |
| The FB makes available the datapoints needed to control the whole device | | |

3.5.1 FB_Counter

3.5.1.1 Functional Block

Channel 0



## SECTION 3: ENERGY MANAGEMENT

3.5.1.2 Description of Communication Objects

| Name in ETPRO 2.10.x | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_PartialResetU32** | **DP_PartialReset** | DPT_Value_4_Ucount (DPT_ID = 12.001) | 1 | W |
| The message is used to reset the partial counter of the configured variable, initialising it to the value contained in the message. | | | | |
| **DPTx_HistoryReset** | **DP_Reset_History** | DPT_Reset (DPT_ID = 1.015) | 2 | W |
| The message is used to reset the entire log stored in the device, i.e. all the measurements saved previously. | | | | |
| **DPTx_FrequencyValue** | **DP_Istantaneous_Consumption** | DPT_Value_Frequency (DPT_ID = 14.033) | 3 | RT |
| Indicates the period between the two most recent samples. | | | | |
| **DPTx_PartialCounter** | **DP_Partial_Counter** | DPT_Value_4_Ucount (DPT_ID = 12.001) | 4 | RT |
| Cumulative value of the counter since the last partial reset control or the last time the device was started. | | | | |
| **DPTx_AbsoluteCounter** | **DP_Absolute_Counter** | DPT_Value_4_Ucount (DPT_ID = 12.001) | 5 | RT |
| Datapoint informing of the change of actuator status. | | | | |
| **DPTx_Alarm** | **DP_Alarm** | DPT_Alarm (DPT_ID = 1.005) | 6 | RT |
| Cumulative value of the counter since the last reset history control. | | | | |
| **DPTx_SysClockDateTime** | **DP_SystemClock** | DPT_DateTime (DPT_ID = 19.001) | 7 | RT |
| The message is received periodically to have the updated time available. | | | | |

# Index

1 Introduction ..... 70
2 Overview of the By-me temperature control system ..... 70
3 Devices 70
3.1 Thermostat (02951) / Temperature probe (20538) ..... 70
3.1.1 FBT_THERMOSTAT ..... 71
3.1.1.1 “Normal” Channel ..... 71
3.1.1.2 Description of Communication Objects ..... 72
3.1.1.3 “Integration” channel ..... 76
3.1.1.4 Description of Communication Objects ..... 77
3.1.2 FBT_OUTPUT ..... 77
3.1.2.1 “On/off valve” channel ..... 77
3.1.2.2 “Proportional valve” channel ..... 77
3.1.2.3 “VIMAR 2T fan coil” channel ..... 77
3.1.2.4 “VIMAR 4T fan coil” channel ..... 78
3.1.2.5 “2T Proportional fan coil” channel ..... 78
3.1.2.6 “4T Proportional fan coil” channel ..... 78
3.1.2.7 Description of Communication Objects ..... 78
3.1.2.8 Description of Communication Objects ..... 78
3.2 HVAC controller (01465) ..... 79
3.2.1 FBT_CONTROLLER ..... 80
3.2.1.1 Functional Block ..... 80
3.2.1.2 Description of Communication Objects ..... 80
3.2.2 FBT_REG_PUMP ..... 82
3.2.2.1 Functional Block ..... 82
3.2.2.2 Description of Communication Objects ..... 82
3.2.3 FBT_REG_HUMIDITY ..... 82
3.2.3.1 Functional Block ..... 82
3.2.3.2 Description of Communication Objects ..... 82
3.2.4 FBT_REG_EXT_TEMP ..... 83
3.2.4.1 Functional Block ..... 83
3.2.4.2 Description of Communication Objects ..... 83
3.2.5 FBT_REG_AUX_TEMP ..... 84
3.2.5.1 “AuxiliaryTemp” channel ..... 84
3.2.5.2 “RegBlockAlarmInput” channel ..... 84
3.3 4-output actuator (01466) ..... 85
3.3.1 FBT_ACTUATOR_PROP ..... 85
3.3.1.1 Functional Block ..... 85
3.3.1.2 Description of Communication Objects ..... 85
3.4 3-input device (1467) ..... 86
3.4.1 FBT_3IN ..... 86
3.4.1.1 “Display” channel ..... 86
3.4.1.2 “Control” channel ..... 86
3.4.1.3 Description of Communication Objects ..... 87
3.4.2 FBT_3IN_VISUAL_TEMP ..... 87
3.4.2.1 Functional block ..... 87
3.4.2.2 Description of Communication Objects ..... 87
3.4.3 FBT_3IN_REG_LUM ..... 88
3.4.2.1 Functional block ..... 88
3.4.2.2 Description of Communication Objects ..... 88
3.5 Eikon Tactil FAN thermostat (21514.F) ..... 90
3.5.1 FBT_THERMOSTAT2 ..... 91
3.5.1.1 “Normal” Channel ..... 91
3.5.1.2 Description of Communication Objects ..... 92
3.5.1.3 “Integration” channel ..... 96
3.5.1.4 Description of Communication Objects ..... 96
3.5.2 FBT_OUTPUT ..... 98
3.5.3 FBT_THERM_PROBE2 ..... 98
3.5.3.1 Functional block ..... 98
3.5.3.2 Description of Communication Objects ..... 98
3.5.4 FBT_THERM_ALARM_TX ..... 98
3.5.4.1 “Window Contact” Channel ..... 98
3.5.4.2 Description of Communication Objects ..... 98
3.5.4.3 “Boiler/heat pump shutdown” CHannel ..... 98
3.5.4.4 Description of Communication Objects ..... 99
3.5.5 FBT_THERM_WINDOW_RX ..... 99
3.5.5.1 Functional block ..... 99
3.5.5.2 Description of Communication Objects ..... 99

3.5.6 FBT_THERM_BLOCK_RX ..... 99
3.5.5.1 Functional block ..... 99
3.5.5.2 Description of Communication Objects ..... 99
3.5.7 FBT_OUTPUT ..... 99
3.6 Eikon Tactil HOTEL thermostat (21514.H) ..... 100
3.6.1 FBT_OUTPUT_SIGNALLING ..... 100
3.6.1.1 Functional block ..... 100
3.6.1.2 Description of Communication Objects ..... 100
3.7 Eikon Tactil STAR thermostat (21514.S) ..... 101
3.7.1 FBT_BUTTON_TOUCH ..... 101
3.8 Eikon Tactil HOTEL thermostat (21514.H) ..... 102
3.8.1 FBT_BUTTON_REDUCED ..... 103
3.8.1.1 “Push button” Channel ..... 103
3.8.1.2 Description of Communication Objects ..... 103
3.8.1.3 “Scenario” Channel ..... 103
3.8.1.4 Description of Communication Objects ..... 103
**4 KNX interfacing ..... 104**
4.1 Mitsubishi via ME-AC-KNX-1-V2 with databank version 0.8 ..... 104
4.2 Mitsubishi via ME-AC-KNX-1-V2/INKNXMIT001I000 with databank version 1.x ..... 105
4.3 Daikin via DK-AC-KNX-1/INKNXDAI001I000 with databank version 0.4 ..... 106
4.4 Daikin via DK-RC-KNX-1/INKNXDAI001R000 with databank version 1.5 ..... 107
4.5 LG via LG-RC-KNX-1i/INKNXLGE001R000 with databank version 1.x ..... 109

# SECTION 4: TEMPERATURE CONTROL

## 1. Introduction

The document describes the features of the KNX communication interfaces of the By-me energy temperature control devices, the method for obtaining the system configuration information and application examples for controlling and commanding system equipment via standard KNX messages.

This document does not describe the functional, technical and regulatory characteristics of the temperature control system, for which you should refer to the official documentation

## 2. Overview of the By-me temperature control system

The By-me devices involved in a temperature control system are:

- Home automation system touch thermostat (02951)
- Temperature probe (20538)
- HVAC controller (01465)
- 4-output actuator (01466)
- 3-input device (01467)
- Eikon Tactil FAN (21514.S), HOTEL (21514.H) and STAR (21514.F) thermostat
- Home automation system dial thermostat (02971)

The other devices perform support functions for managing the climate zones. For example, the 3-input device (01467) can provide the external temperature to the HVAC controller device (01465) to calculate the delivery water temperature, while the 4-output actuator (01466) can be controlled by the thermostat (02951) to switch on a radiator.

For more information on the technical characteristics and installation techniques, please refer to the manual of the Control Unit 21509 - Temperature Control Section.

## 3. Devices

### 3.1 Thermostat (02951) / Temperature probe (20538)

The devices consist of 5 Functional Blocks:

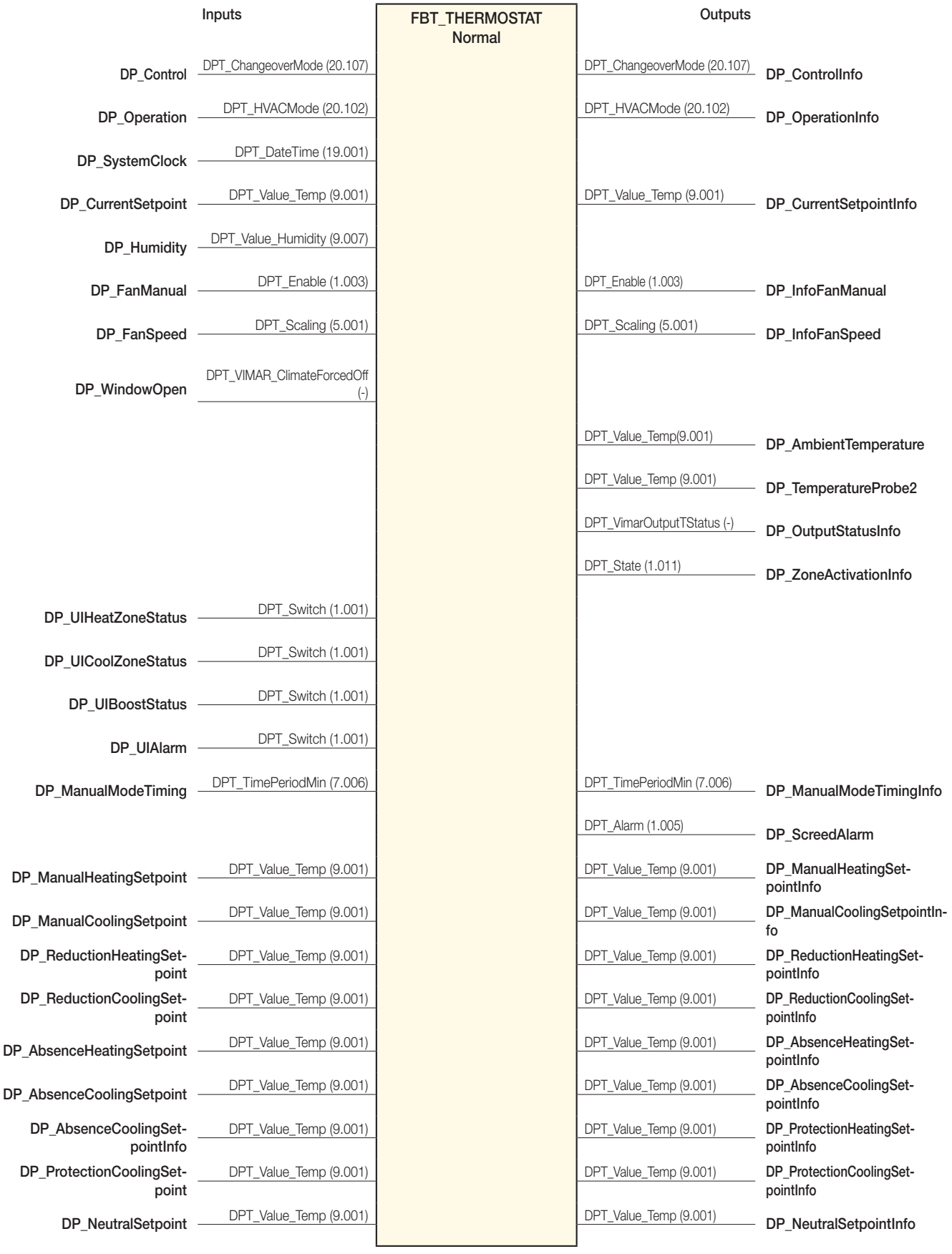
| Fb Index | Fb Name | Fb Type |
|---|---|---|
| 1 | FB_Thermostat | FBT_THERMOSTAT (0x5B) |
| Thermostat | | |
| 2 | FB_Output1 | FBT_OUTPUT (0x5C) |
| Output 1 control | | |
| 3 | FB_Output2 | FBT_OUTPUT (0x5C) |
| Output 2 control | | |
| 4 | FB_Output3 | FBT_OUTPUT (0x5C) |
| Output 3 control | | |
| 5 | FB_Output4 | FBT_OUTPUT (0x5C) |
| Output 4 control | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.1.1 FB_THERMOSTAT

#### 3.1.1.1 “Normal” Channel

##### 3.1.1.1.1 Functional Block



## SECTION 4: TEMPERATURE CONTROL

### 3.1.1.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ChangeOverMode** | **DP_Control** | DPT_ChangeoverMode(20.107) | 1 | W |
| Command for changing control mode:<br>0 = neutral zone (auto)<br>1 = air-conditioning (cooling only)<br>2 = heating (heating only)<br><br>* in integration mode (PAR_AdjustableSelect=Integration) it controls the "antifreeze" or "too hot" display icon<br>* in the neutral zone it is used to limit the operating mode<br>* on actually changing the operating mode, a message is generated by DP_ControlInfo | | | | |
| **DPTx_ChangeOverModeInfo** | **DP_ControlInfo** | DPT_ChangeoverMode(20.107) | 2 | RT |
| Provides the current status of the control mode.<br>It is sent on changing the mode. | | | | |
| **DPTx_HvacMode** | **DP_Operation** | DPT_HVACMode(20.102) | 3 | W |
| Command for changing operating mode:<br>KNX - VIMAR<br>(0) auto = auto<br>(1) comfort = manual<br>(2) standby = reduction<br>(3) economy = absence<br>(4) building protection = protection (anti-freez/too hot)<br>(5) / = manual, with timer<br>(6) / = off<br><br>* Changes the operating mode subject to the limitations on use and constraints of changing mode<br>* In integration mode it is used to access the respective icons on the thermostat display and to receive commands from a viewer and forward them to the third-party system via the corresponding Info DP | | | | |
| **DPTx_HvacModeInfo** | **DP_OperationInfo** | DPT_HVACMode(20.102) | 4 | RT |
| Provides the current status of the operating mode.<br>It is sent on changing the mode.<br><br>* In integration it is used to control the third-party system in the operating mode chosen on the thermostat or received from the display | | | | |
| **DPTx_SysClockDateTime** | **DP_SystemClock** | DPT_DateTime(19.001) | 5 | W |
| Updates the device clock.<br>Used for the time schedule.<br>When the device is started it asks the system for the time, if this does not arrive it works with its own internal clock. | | | | |
| **DPTx_TemperatureSetpoint1** | **DP_CurrentSetpoint** | DPT_Value_Temp(9.001) | 6 | W |
| Used to set the current setpoint value. This value is normally propagated to the relative DP_Setpoint* of the current operating mode, in some cases this does not occur. Read the specifications. | | | | |
| **DPTx_TemperatureSetpointInfo1** | **DP_CurrentSetpointInfo** | DPT_Value_Temp(9.001) | 7 | RT |
| Provides the current setpoint value.<br>It is sent upon a change.<br><br>So it is also sent when the control mode or operating mode change. | | | | |
| **DPTx_HumidityValue *** | **DP_Humidity** | DPT_Value_Humidity(9.007) | 8 | W |
| Percentage value of relative humidity, to be shown on the thermostat display.<br>It is sent from the humidity sensor, via the analogue inputs. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_FanManualEnable** | **DP_FanManual** | DPT_Enable(1.003) | 9 | W |
| Indicates the type of fan coil fan control:<br>0 = automatic<br>1 = manual<br><br>in the case of manual speed, it uses the value in DP_FanSpeed to control the speed of the fans. | | | | |
| **DPTx_FanManualEnableInfo** | **DP_InfoFanManual** | DPT_Enable(1.003) | 10 | RT |
| Provides the status of the relevant DP.<br>It is sent upon a change. | | | | |
| **DPTx_FanSpeed** | **DP_FanSpeed** | DPT_Scaling(5.001) | 11 | W |
| For the manual control of the fan coil fans, it indicates the value to control the fan speed at.<br>In the case of a 3-speed fan coil:<br>[0%,33%] => V1<br>[33%, 67%] => V2<br>[67%, 100%] => V3<br>CAUTION: 0% (fans off, OFF) switching off the fans is not allowed<br><br>On the contrary, in integration mode the value of 0% may change the graphic symbol to OFF, indicating that the fans have been switched off by the third-party system. | | | | |
| **DPTx_FanSpeedInfo** | **DP_InfoFanSpeed** | DPT_Scaling(5.001) | 12 | RT |
| Provides the current status of the fan speed.<br>It is sent upon a change.<br>If there is no fan coil as an output device or the fans are off it returns to 0%.<br><br>In the case of a 3-speed fan coil, the DP admits the following values:<br>V1 = 33% (0x55)<br>V2 = 67% (0xAA)<br>V3 = 100% (0xFF) | | | | |
| **DPTx_ClimateForcedOff** | **DP_WindowOpen** | DPT_VIMAR_ClimateForcedOff(-) | 13 | W |
| Signals the condition of an open window<br><br>Initially the idea was to use DPT_Window_Door (1019), but later it was noticed that for backward compatibility with the existing contact interfaces it is advisable to use the previous VIMAR standard DPT.<br><br>In "integration" mode it is present and used to connect a dry contact to a third-party system, this is why the DP must place the system in OFF and restore it following the relative parameters.<br><br>Format:<br>0 = open window (request for forcing the system OFF)<br>1 = window closed | | | | |
| **DPTx_AmbientTemperature** | **DP_AmbientTemperature** | DPT_Value_Temp(9.001) | 14 | RT |
| Provides the temperature value used by the device to perform temperature control.<br>This value can be composed of:<br>- Probe 1<br>- Probe 2<br>- Average between Probe 1 and Probe 2<br><br>It is sent with a variation of 0.1°C<br><br>In integration mode it serves to provide third parties with the temperature value measured by the probe. | | | | |
| **DPTx_TemperatureValue** | **DP_TemperatureProbe2** | DPT_Value_Temp(9.001) | 15 | RT |
| Provides the temperature value measured by Probe 2, in any condition configured.<br><br>The typical use is when PAR_UtilizzoSonda2 = 4 (for display only).<br>In other cases it is not used in the system.<br><br>It is sent with a variation of 0.1°C<br><br>In integration mode it serves to provide third parties with the temperature value measured by the probe. | | | | |

## SECTION 4: TEMPERATURE CONTROL

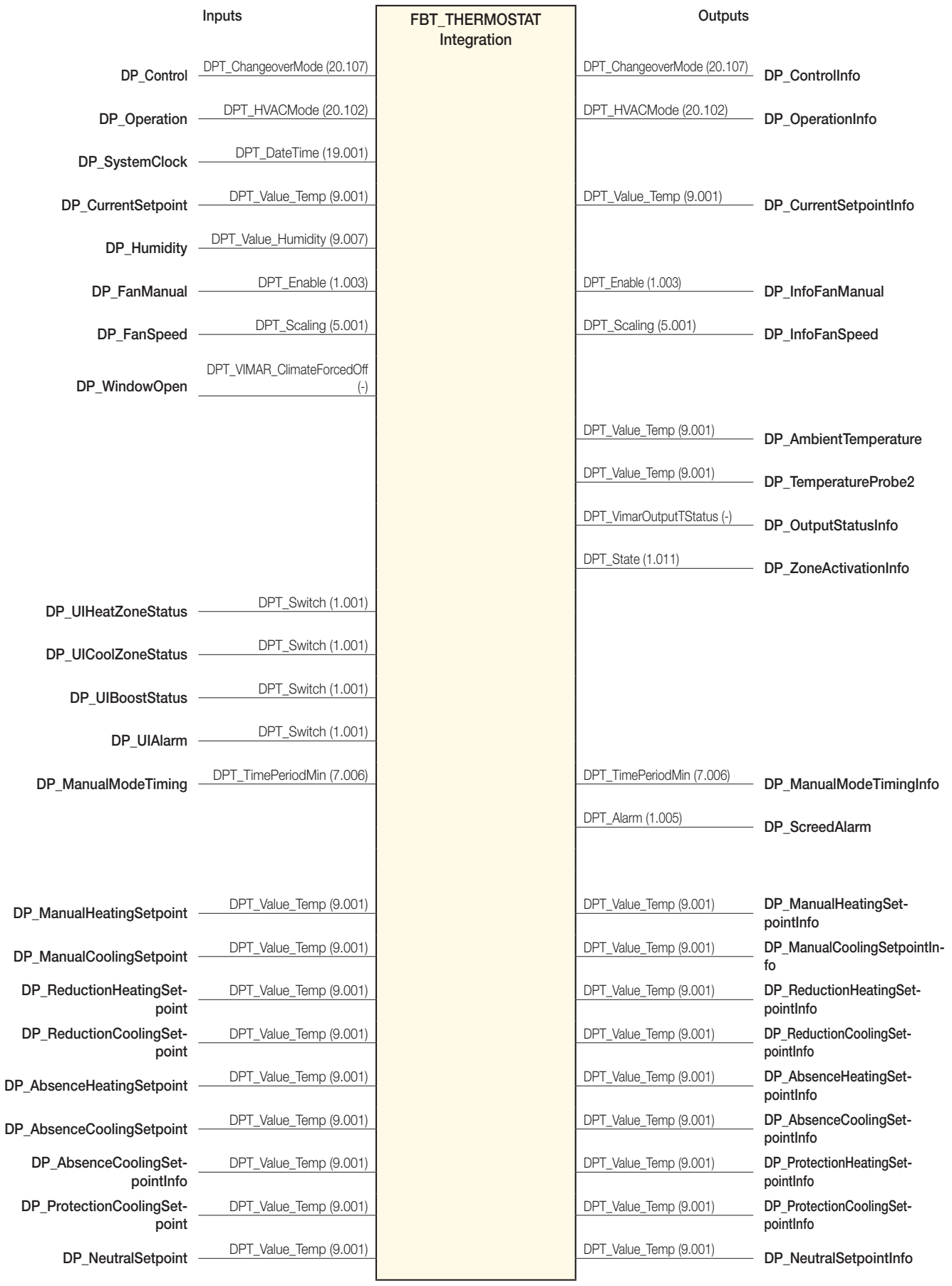
<table>
<tr><th>Name in VIEW Pro</th><th>Datapoint</th><th>DPT</th><th>CO number</th><th>Flags</th></tr>
<tr><td>DPTx_OutStatus</td><td>DP_OutputStatusInfo</td><td>DPT_VimarOutputTStatus(-)</td><td>16</td><td>RT</td></tr>
<tr><td colspan="5">Provides the current status of all the thermostat outputs. Only on/off status.<br>It is used for supervisors, to have summary information without checking all the outputs.<br><br>Basically it is a representation of the flame, snowflake and star icons on the thermostat GUI.<br><br>It is sent upon a change.<br><br>The format:<br><br>DPT_State[3] (1.011)<br><br>Format: 1 octet: B3<br>octet no: 1<br><br><table><tr><td></td><td></td><td></td><td></td><td></td><td>Main Heat.</td><td>Main Cool.</td><td>Boost</td></tr></table><br>0 = off<br>1 = on</td></tr>
<tr><td>DPTx_ManualModeTiming</td><td>DP_ManualModeTiming</td><td>DPT_TimePeriodMin(7.006)</td><td>22</td><td>W</td></tr>
<tr><td colspan="5">Value used by the thermostat during the "timed manual" operating mode.<br>When changing the mode to "timed manual", the timer is started with a value equal to this DP.<br>During "timed manual" mode, a change in this DP restarts the timer on the new value.<br>At the end of the set time, the operating mode returns to the previous one.<br><br>Value not permitted: 0<br><br>Example:<br>By default it starts at 1h, then an update will come from the DP which will restart the timer and restart the count. The next time it will start with the last value received.</td></tr>
<tr><td>DPTx_ManualModeTimingInfo</td><td>DP_ManualModeTimingInfo</td><td>DPT_TimePeriodMin(7.006)</td><td>23</td><td>RT</td></tr>
<tr><td colspan="5">Provides the value of the relevant DP.<br>It is sent upon a change.</td></tr>
<tr><td>DPTx_SubFloorAlarm</td><td>DP_ScreedAlarm</td><td>DPT_Alarm(1.005)</td><td>24</td><td>RT</td></tr>
<tr><td colspan="5">It is generated in the event of a screed limitation alarm.<br>It remains set on 1 until the alarm ends (interaction by the user is needed to reset the alarm)</td></tr>
<tr><td>DPTx_ManualHeatingSetpoint</td><td>DP_ManualHeatingSetpoint</td><td>DPT_Value_Temp(9.001)</td><td>26</td><td>W</td></tr>
<tr><td colspan="5">Direct access to the setpoint relating to the operation and control mode indicated by the DP name.</td></tr>
<tr><td>DPTx_ManualHeatingSetpointInfo</td><td>DP_ManualHeatingSetpointInfo</td><td>DPT_Value_Temp(9.001)</td><td>27</td><td>RT</td></tr>
<tr><td colspan="5">Value of the setpoint relating to the operation and control mode indicated by the DP name.<br><br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*)</td></tr>
<tr><td>DPTx_ManualCoolingSetpoint</td><td>DP_ManualCoolingSetpoint</td><td>DPT_Value_Temp(9.001)</td><td>28</td><td>W</td></tr>
<tr><td colspan="5">Direct access to the setpoint relating to the operation and control mode indicated by the DP name.</td></tr>
<tr><td>DPTx_ManualCoolingSetpointInfo</td><td>DP_ManualCoolingSetpointInfo</td><td>DPT_Value_Temp(9.001)</td><td>29</td><td>RT</td></tr>
<tr><td colspan="5">Value of the setpoint relating to the operation and control mode indicated by the DP name.<br><br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*)</td></tr>
</table>

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ReductionHeatingSetpoint** | **DP_ReductionHeatingSetpoint** | DPT_Value_Temp(9.001) | 30 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ReductionHeatingSetpointInfo** | **DP_ReductionHeatingSetpointInfo** | DPT_Value_Temp(9.001) | 31 | RT |
| Value of the setpoint relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_ReductionCoolingSetpoint** | **DP_ReductionCoolingSetpoint** | DPT_Value_Temp(9.001) | 32 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ReductionCoolingSetpointInfo** | **DP_ReductionCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 33 | RT |
| Value of the setpoint relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_AbsenceHeatingSetpoint** | **DP_AbsenceHeatingSetpoint** | DPT_Value_Temp(9.001) | 34 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_AbsenceHeatingSetpointInfo** | **DP_AbsenceHeatingSetpointInfo** | DPT_Value_Temp(9.001) | 35 | RT |
| Value of the setpoint relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_AbsenceCoolingSetpoint** | **DP_AbsenceCoolingSetpoint** | DPT_Value_Temp(9.001) | 36 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_AbsenceCoolingSetpointInfo** | **DP_AbsenceCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 37 | RT |
| Value of the setpoint relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_AbsenceCoolingSetpointInfo** | **DP_AbsenceCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 38 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ProtectionHeatingSetpointInfo** | **DP_ProtectionHeatingSetpointInfo** | DPT_Value_Temp(9.001) | 39 | RT |
| Value of the setpoint relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_ProtectionCoolingSetpoint** | **DP_ProtectionCoolingSetpoint** | DPT_Value_Temp(9.001) | 40 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ProtectionCoolingSetpointInfo** | **DP_ProtectionCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 41 | RT |
| Value of the setpoint relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_NeutralSetpoint** | **DP_NeutralSetpoint** | DPT_Value_Temp(9.001) | 42 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_NeutralSetpointInfo** | **DP_NeutralSetpointInfo** | DPT_Value_Temp(9.001) | 43 | RT |
| Value of the setpoint relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.1.1.3 “Integration” Channel



## SECTION 4: TEMPERATURE CONTROL

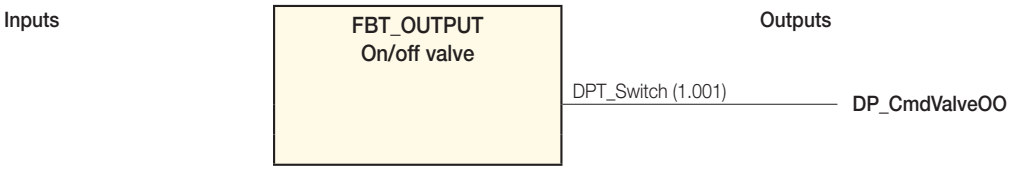
#### 3.1.1.3.1 Description of Communication Objects

The "Integration" channel provides for the following Communication Objects, *in addition to those already provided for by the "normal" channel.*

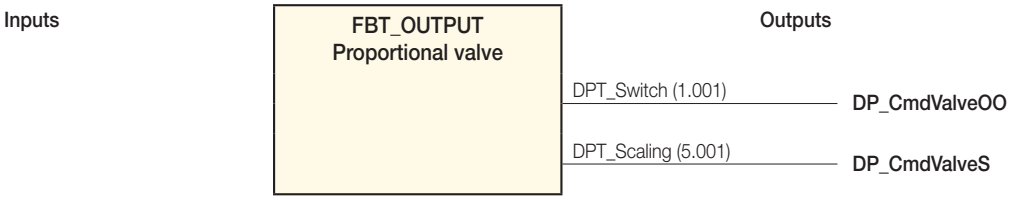
| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ZoneActivationInfo** | **DP_ZoneActivationInfo** | DPT_State(1.011) | 17 | RT |
| Provides the status of the zone. That is, it informs if the thermostat is on and not off.<br>It is used for third-party integration to understand, with a single DP, when the system must be switched on or off.<br>For example, this DP is NOT used by the Circulation Pump device to understand when to switch on, because it provides no information on the status of the Heat/Cool output.<br>It is sent upon a change.<br>0 = zone off<br>1 = if DP_OperationInfo!= 6 (OFF) | | | | |
| **DPTx_HeatingStatus** | **DP_UIHeatZoneStatus** | DPT_Switch(1.001) | 18 | W |
| Used in integration for managing the "flame" icon.<br>It updates the DP_OutputStatusInfo. | | | | |
| **DPTx_CoolingStatus** | **DP_UICoolZoneStatus** | DPT_Switch(1.001) | 19 | W |
| Used in integration for managing the "snowflake" icon.<br>It updates the DP_OutputStatusInfo. | | | | |
| **DPTx_BoostStatus** | **DP_UIBoostStatus** | DPT_Switch(1.001) | 20 | W |
| Used in integration for managing the "boost" icon.<br>It updates the DP_OutputStatusInfo. | | | | |
| **DPTx_UIAlarm** | **DP_UIAlarm** | DPT_Switch(1.001) | 21 | W |
| Used in integration for managing the "alarm" icon. Icon control is only available if screed limitation is not enabled in order to avoid conflicts of use. | | | | |

### 3.1.2 FBT_OUTPUT

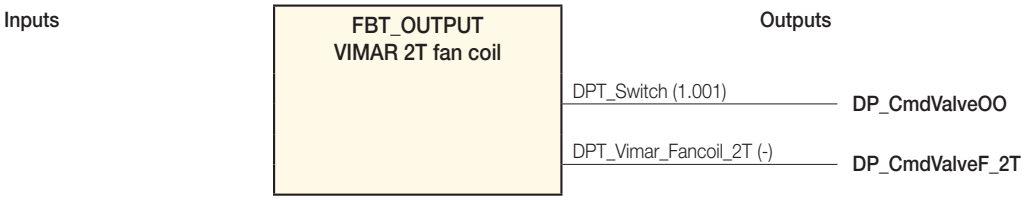
#### 3.1.2.1 "On/off valve" channel




#### 3.1.2.2 "Proportional valve" channel

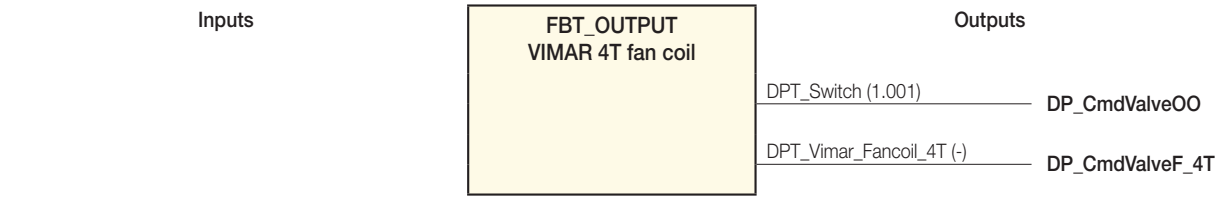



#### 3.1.2.3 "VIMAR 2T fan coil" channel

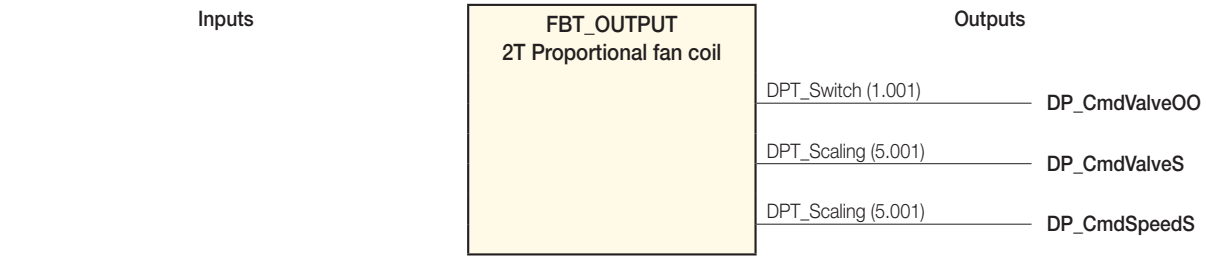


## SECTION 4: TEMPERATURE CONTROL

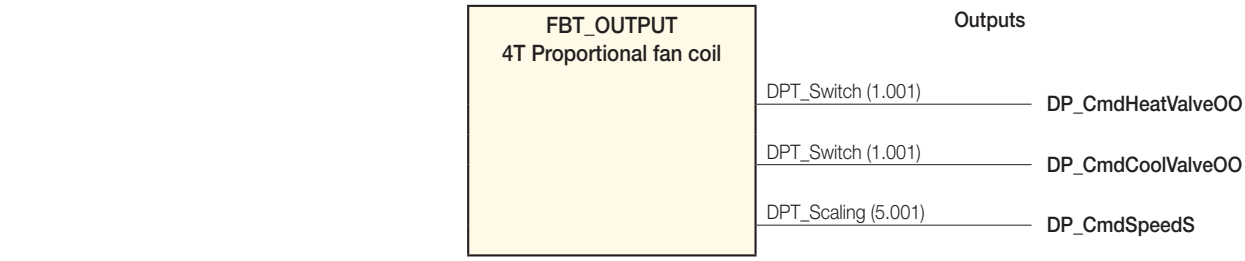
### 3.1.2.4 "VIMAR 4T fan coil" channel




### 3.1.2.5 "2T proportional fan coil" channel




### 3.1.2.6 "4T proportional fan coil" channel




### 3.1.2.7 Description of Communication Objects

Detailed descriptions of each Datapoint covered by the model in the various channels are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_CmdValveOO** | DPT_Switch(1.001) | 1 | T |
| On/off valve command.<br>It is also sent on commands of different types of valves (proportional, fan coil) to inform the system whether they are open or closed. That is, in the proportional case if 0% → OFF, otherwise ON<br>It is also used for being connected to the circulation pump device | | | | |
| **DPTx_Value** | **DP_CmdValveS** | DPT_Scaling(5.001) | 2 | T |
| Proportional valve command.<br>0% = valve off | | | | |
| **DPTx_Fancoil2Pipes** | **DP_CmdValveF_2T** | DPT_Vimar_Fancoil_2T(-) | 3 | T |

Vimar-type fancoil command to manage 2-pipe fan coils.
Managed paired with device 1851 (4 relays) in which 3 relays are used for the speed, the other to manage the valve which can be for heating or air-conditioning depending on the temperature of the water that is set upstream.

Format:

DPT_Vimar_Fancoil_2T: 1 byte

| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Valv. Heat/Cool | | | | | Speed 3 | Speed 2 | Speed 1 |

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_Fancoil4Pipes** | **DP_CmdValveF_4T** | DPT_Vimar_Fancoil_4T(-) | 4 | T |

Vimar-type fancoil command to manage 4-pipe fan coils.
Managed paired with devices 1850 (1 relay) + 1851 (4 relays) in which 3 relays are used for the speed, the other two relays are used to control the hot valve and the cold valve.

Format:

DPT_Vimar_Fancoil_4T: 1 byte

| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Valv. Heat. | Valv. Cool. | | | | Speed 3 | Speed 2 | Speed 1 |

| | | | | |
|---|---|---|---|---|
| **DPTx_AbsoluteSpeed** | **DP_CmdSpeedS** | DPT_Scaling(5.001) | 8 | T |

In the case of a proportional fan coil
Requires operation at the speed described in the parameter name
0% = valves off

| | | | | |
|---|---|---|---|---|
| **DPTx_SwitchHot** | **DP_CmdHeatValveOO** | DPT_Switch(1.001) | 9 | T |

Heating on/off valve command

| | | | | |
|---|---|---|---|---|
| **DPTx_SwitchCold** | **DP_CmdCoolValveOO** | DPT_Switch(1.001) | 10 | T |

Cooling on/off valve command

### 3.2 HVAC controller (01465)

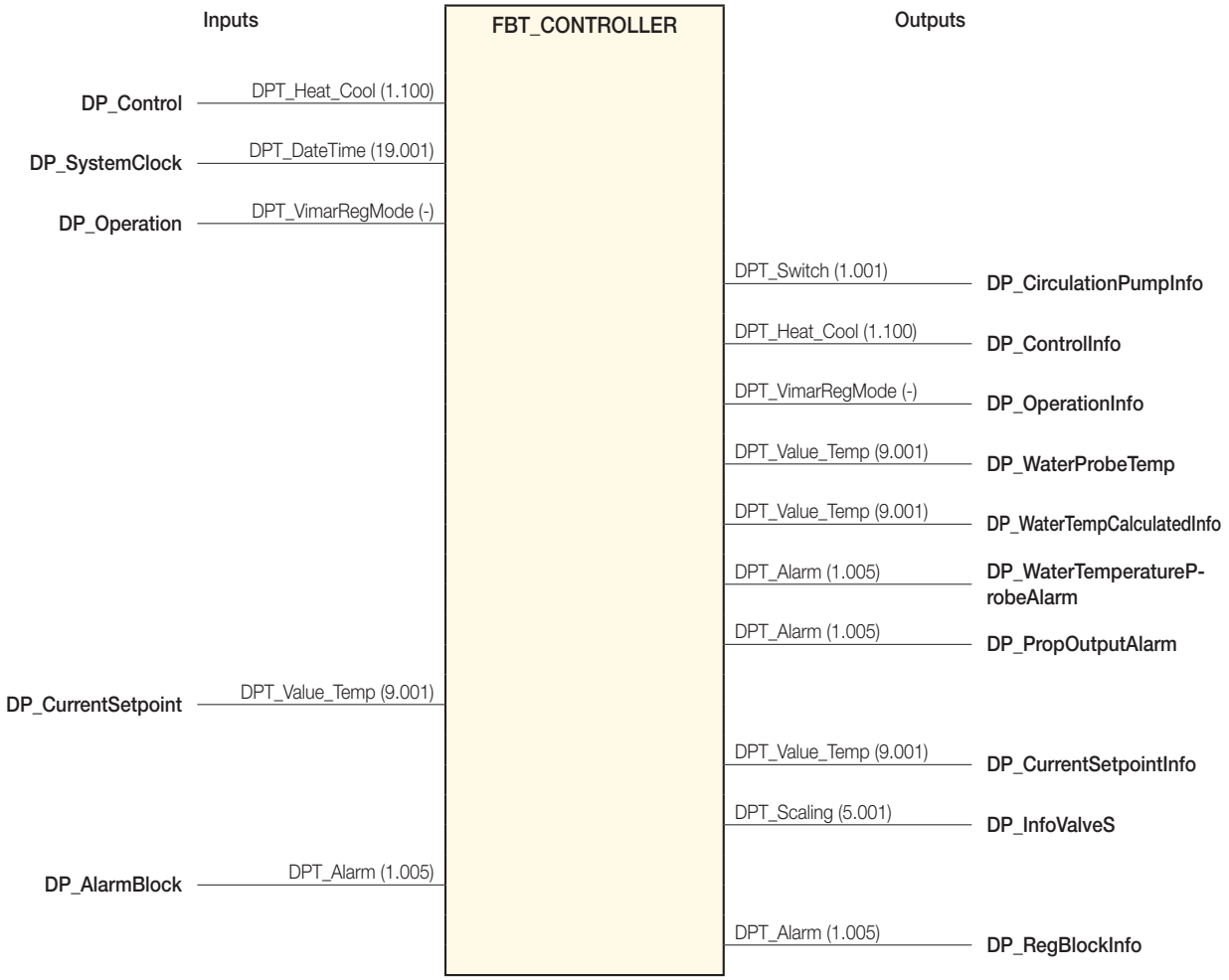
The device contains the following FB, allocated in the position indicated in the table:

| Fb Index | Fb Name | Fb Type |
|---|---|---|
| **1** | FB_Controller | FBT_CONTROLLER (0x5D) |
| HVAC controller | | |
| **2** | FB_CirculationPump | FBT_REG_PUMP (0x5E) |
| Circulation pump | | |
| **3** | FB_Humidity | FBT_REG_HUMIDITY (0x5F) |
| Humidity controller | | |
| **4** | FB_ExternalTemp | FBT_REG_TEMP_EXT (0x60) |
| External temperature controller | | |
| **5** | FB_AuxiliaryTemp | FBT_REG_TEMP_AUX (0x61) |
| Auxiliary temperature controller | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.2.1 FBT_CONTROLLER

#### 3.2.1.1 Functional Block




#### 3.2.1.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_HeatCoolMode** | **DP_Control** | DPT_Heat_Cool(1.100) | 1 | W |
| Control mode change command:<br>0 = air-conditioning (cooling)<br>1 = heating (heating) | | | | |
| **DPTx_SysClockDateTime** | **DP_SystemClock** | DPT_DateTime(19.001) | 2 | W |
| Updates the device clock.<br>Used for the time schedule. | | | | |
| **DPTx_SubsetHvacMode** | **DP_Operation** | DPT_VimarRegMode(-) | 3 | W |
| Operating mode.<br>Format:<br>0 = auto<br>1 = comfort<br>2 = economy<br>3 = off | | | | |

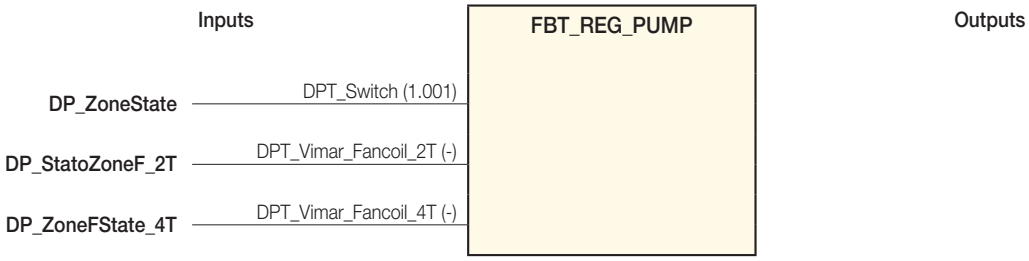
## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo** | **DP_CirculationPumpInfo** | DPT_Switch(1.001) | 4 | RT |
| Info on the output status of the circulation pump.<br>It informs the system on the status of the circulation pump relay.<br>Basically, it is the output of the OR of all the climate connected to the controller | | | | |
| **DPTx_HeatCoolModeInfo** | **DP_ControlInfo** | DPT_Heat_Cool(1.100) | 5 | RT |
| Provides the current status of the control mode.<br>It is sent on changing the mode. | | | | |
| **DPTx_SubsetHvacModeInfo** | **DP_OperationInfo** | DPT_VimarRegMode(-) | 6 | RT |
| Provides the current status of the operating mode.<br>It is sent on changing the mode. | | | | |
| **DPTx_WaterTemperature** | **DP_WaterProbeTemp** | DPT_Value_Temp(9.001) | 7 | RT |
| Temperature value read by the hot water delivery probe.<br>It is sent with a variation of 0.1°C | | | | |
| **DPTx_WaterTemperatureCalculated** | **DP_WaterTempCalculatedInfo** | DPT_Value_Temp(9.001) | 8 | RT |
| Value of the delivery temperature calculated by the controller.<br>It is sent with a variation of 0.1°C | | | | |
| **DPTx_WaterTemperatureProbeAlarm** | **DP_WaterTemperatureProbeAlarm** | DPT_Alarm(1.005) | 9 | RT |
| Alarm signal. Sent in the case of a status change.<br>Sent in the case of failure of the delivery probe. | | | | |
| **DPTx_OutputAlarm** | **DP_PropOutputAlarm** | DPT_Alarm(1.005) | 10 | RT |
| Alarm signal. Sent in the case of a status change.<br>Sent in the case of overload of the proportional output | | | | |
| **DPTx_TemperatureSetpoint1** | **DP_CurrentSetpoint** | DPT_Value_Temp(9.001) | 11 | W |
| Used to set the current setpoint value. This value is normally propagated to the relevant parameter (PAR_SpAmb*, PAR_SpDeliv*).<br>In the case of a time schedule (auto operation) writing to this DP is ignored.<br>Permitted values: depend on the Sp referred to<br>(see constraints Par_Sp*) | | | | |
| **DPTx_TemperatureSetpointInfo1** | **DP_CurrentSetpointInfo** | DPT_Value_Temp(9.001) | 12 | RT |
| Provides the current setpoint value.<br>It is sent upon a change.<br>So it is also sent when the control mode or operating mode change. | | | | |
| **DPTx_ValueInfo** | **DP_InfoValveS** | DPT_Scaling(5.001) | 13 | RT |
| Returns the value of the mixing valve opening percentage.<br>0% = valve off<br>Sent on a change | | | | |
| **DPTx_ClimateRegulatorArrest1** | **DP_AlarmBlock** | DPT_Alarm(1.005) | 15 | W |
| In the event of an alarm (DPT_Alarm = 1 (Alarm)) HVAC controller shutdown is activated and then both the pump and the valve are closed.<br>At the same time this generates sending the status change from DP_RegBlockInfo<br>When the alarm ends (DPT_Alarm = 0 (no alarm)), the controller starts working as before and updates the DP_RegBlockInfo | | | | |
| **DPTx_ClimateRegulatorArrestInfo** | **DP_RegBlockInfo** | DPT_Alarm(1.005) | 16 | RT |
| Notifies the change in status of the controller in the event of its shutdown caused by DP_AlarmBlock or by the dry contact input of FB_AuxiliaryTemp.<br>Sent on a change | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.2.2 FBT_REG_PUMP

#### 3.2.2.1 Functional Block



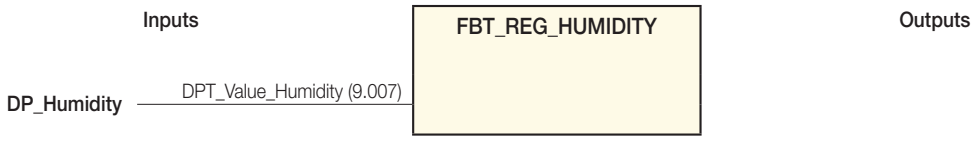

#### 3.2.2.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_ZoneState** | DPT_Switch(1.001) | 1 | W |
| Monitors up to 128 climate zones and calculates the OR to change the status of the relay connected to the circulation pump.<br>It is the task of the configurator to limit the maximum number of zones to be monitored, to date the maximum limit is 64.<br>The change causes the DP_CirculationPumpInfo to be sent from FB_Controller | | | | |
| **DPTx_Fancoil2Pipes** | **DP_StatoZoneF_2T** | DPT_Vimar_Fancoil_2T(-) | 2 | W |
| Like DP_ZoneState but referred to 2-pipe fan coil devices. | | | | |
| **DPTx_Fancoil4Pipes** | **DP_ZoneFState_4T** | DPT_Vimar_Fancoil_4T(-) | 3 | W |
| Like DP_ZoneState but referred to 4-pipe fan coil devices.<br>Depending on the current PAR_WorkMode* the device controls the bit relating to heating or air-conditioning. Therefore it does not require a dedicated parameter that instead was present in the old device 01850<br>(1 Relay for DIN rail with circulation pump function). | | | | |

### 3.2.3 FBT_REG_HUMIDITY

#### 3.2.3.1 Functional Block




#### 3.2.3.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

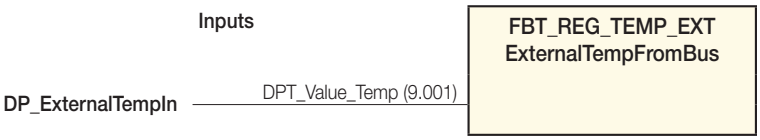
| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_HumidityValue** | **DP_Humidity** | DPT_Value_Humidity(9.007) | 1 | W |
| Relative humidity of the entire system for the calculation of the minimum delivery in the case of air-conditioning to prevent condensation. | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.2.4 FBT_REG_TEMP_EXT

#### 3.2.4.1 “ExternalTempFromBus” channel

##### 3.2.4.1.1 Functional Block



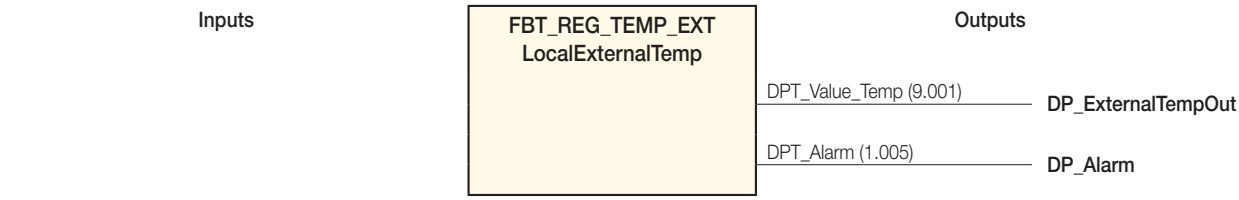

##### 3.2.4.1.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_TemperatureValue** | **DP_ExternalTempIn** | DPT_Value_Temp(9.001) | 1 | W |
| Temperature read by bus. No temperature probe is connected to the device terminals. | | | | |

#### 3.2.4.2 “LocalExternalTemp” channel

##### 3.2.4.2.1 Functional Block




##### 3.2.4.2.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

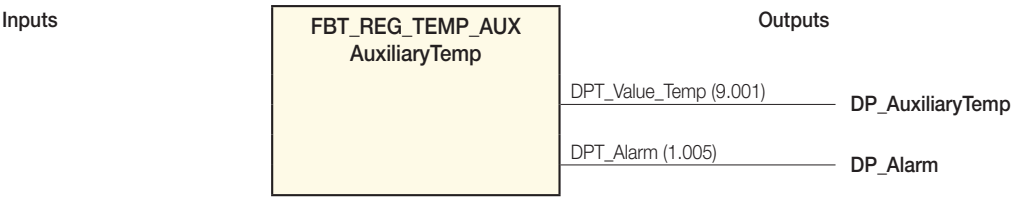
| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_TemperatureValue** | **DP_ExternalTempOut** | DPT_Value_Temp(9.001) | 2 | RT |
| Temperature read by the probe connected to the device terminals. | | | | |
| **DPTx_ExternalTemperatureProbeAlarm** | **DP_Alarm** | DPT_Alarm(1.005) | 3 | RT |
| Alarm signal. Sent in the case of a status change.<br>Sent in the case of probe failure (short or open). | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.2.5 FBT_REG_TEMP_AUX

#### 3.2.5.1 “AuxiliaryTemp” channel

##### 3.2.5.1.1 Functional Block




##### 3.2.5.1.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_TemperatureValue** | **DP_AuxiliaryTemp** | DPT_Value_Temp(9.001) | 1 | RT |
| Auxiliary temperature read by the probe connected to the relative device terminal. | | | | |
| **DPTx_AuxiliaryTemperatureProbeAlarm** | **DP_Alarm** | DPT_Alarm(1.005) | 3 | RT |
| Alarm signal. Sent in the case of a status change.<br>Sent in the case of probe failure | | | | |

#### 3.2.5.2 “RegBlockAlarmInput” channel

##### 3.2.5.2.1 Functional Block

##### 3.2.5.2.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ClimateRegulatorArrest2** | **DP_ContactAlarmInfo** | DPT_Alarm(1.005) | 2 | RT |
| Interprets the value from the auxiliary probe input as a dry contact input (interpreted with the meaning of PAR_Polarity) and sends its value via this DP.<br>Used to notify a blocking alarm on FB_Controller (this is done through a link inside the FB).<br>Also in this case this information is made available on the bus. | | | | |

## SECTION 4: TEMPERATURE CONTROL

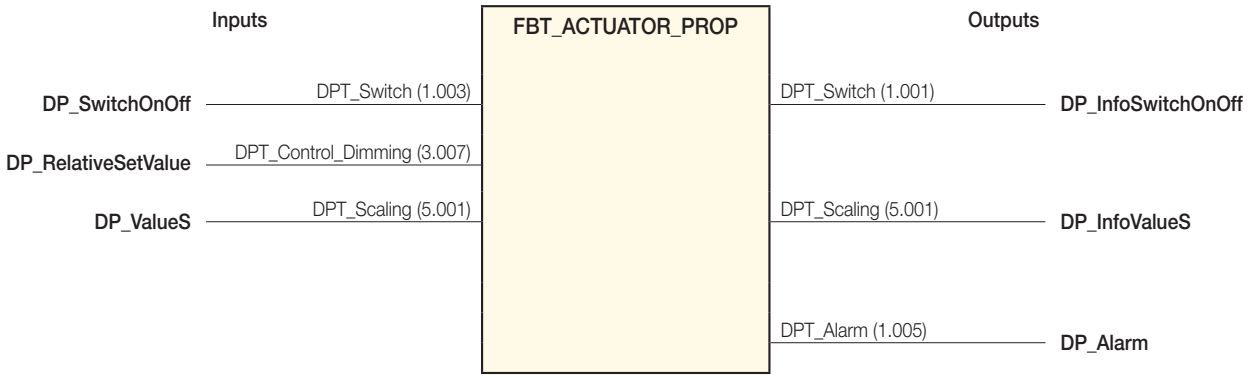
### 3.3 4-output actuators (01466)

The device contains the following FB, allocated in the position indicated in the table:

| Fb Index | Fb Name | Fb Type |
|---|---|---|
| 1 | FB_ACTUATOR_PROP_1 | FBT_ACTUATOR_PROP (0x5A) |
| Output 1 | | |
| 2 | FB_ACTUATOR_PROP_2 | FBT_ACTUATOR_PROP (0x5A) |
| Output 2 | | |
| 3 | FB_ACTUATOR_PROP_3 | FBT_ACTUATOR_PROP (0x5A) |
| Output 3 | | |
| 4 | FB_ACTUATOR_PROP_4 | FBT_ACTUATOR_PROP (0x5A) |
| Output 4 | | |

#### 3.3.1 FBT_ACTUATOR_PROP

##### 3.3.1.1 Functional Block




##### 3.3.1.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_SwitchOnOff** | DPT_Switch(1.003) | 1 | W |
| Actuator on and off control. | | | | |
| **DPTx_ControlValue** | **DP_RelativeSetValue** | DPT_Control_Dimming(3.007) | 2 | W |
| Control for regulating the value of the actuator output channel. | | | | |
| **DPTx_Value** | **DP_ValueS** | DPT_Scaling (5.001) | 3 | W |
| Percentage value of the actuator output channel. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoSwitchOnOff** | DPT_Switch (1.001) | 5 | RT |
| Current value of the status of the actuator. | | | | |
| **DPTx_ValueInfo** | **DP_InfoValueS** | DPT_Scaling (5.001) | 6 | RT |
| Current value of the adjustment percentage of the output channel. | | | | |
| **DPTx_Alarm** | **DP_Alarm** | DPT_Alarm (DPT_ID = 1.005) | 7 | RT |
| Output channel alarm status signal. | | | | |

N.B. Configured in the output with the proportional valve of a thermostat: value -> Absolute speed, Alarm -> Alarm

## SECTION 4: TEMPERATURE CONTROL

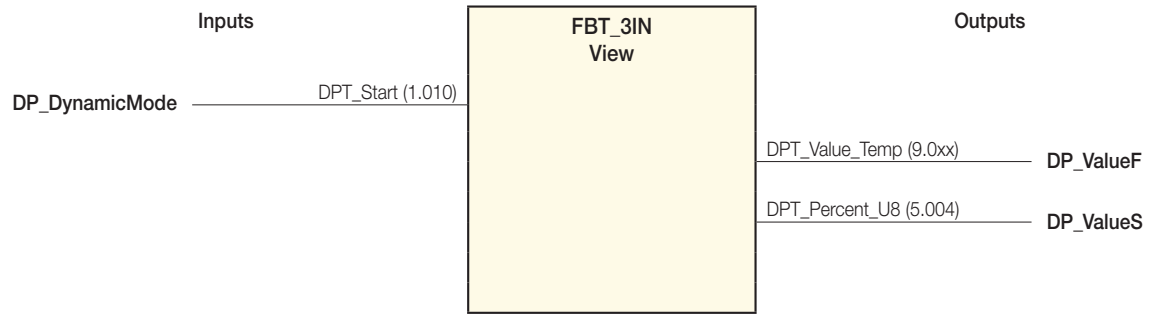
### 3.4 3-input device (1467)

The device contains the following FB, allocated in the position indicated in the table:

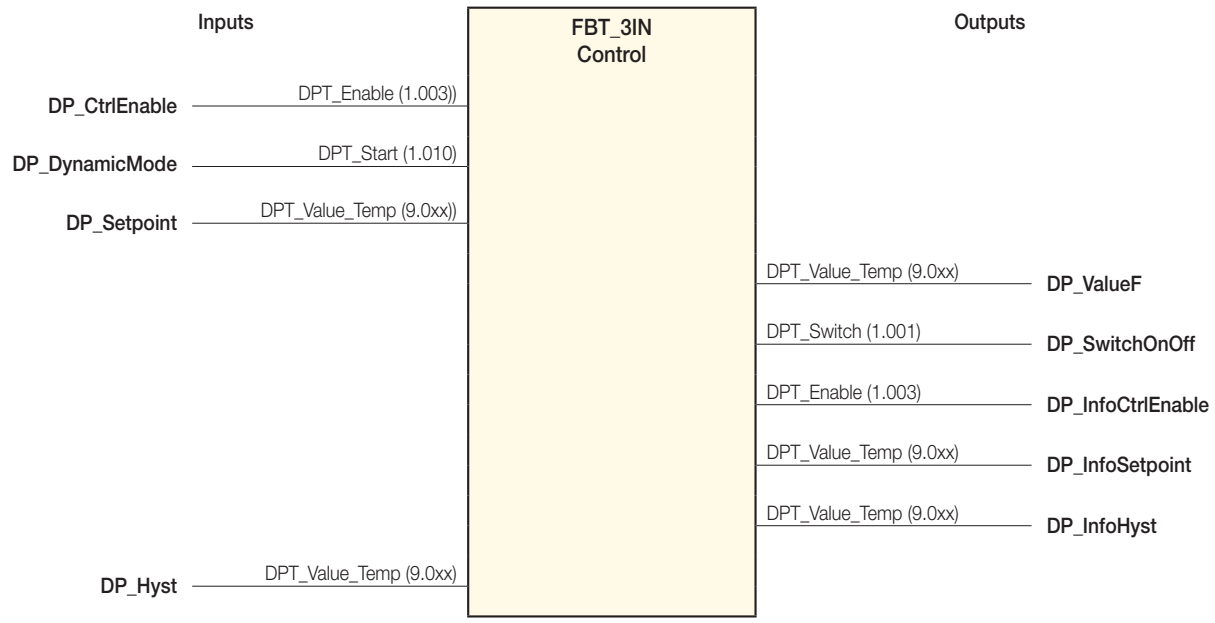
| Fb Index | Fb Name | Fb Type |
|---|---|---|
| 1 | FB_3IN | FBT_3IN (0x57) |
| Generic input for voltage/current sensors | | |
| 2 | FB_3IN_VISUAL_TEMP | FBT_3IN_VISUAL_TEMP (0x58) |
| Temperature input for NTC probe | | |
| 3 | FB_3IN_REG_LUM | FBT_3IN_REG_LUM (0x59) |
| Dimmer | | |

#### 3.4.1 FBT_3IN

##### 3.4.1.1 “Display” channel




##### 3.4.1.2 “Control” channel




**Note:** The above example is configured as temperature; all the dptx with * vary according to the configured value.

## SECTION 4: TEMPERATURE CONTROL

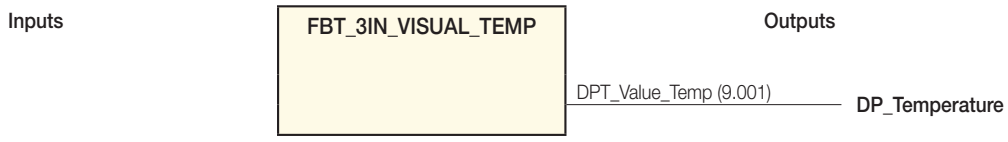
#### 3.4.1.3 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model in the 2 channels are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_Enable** | **DP_CtrlEnable** | DPT_Enable(1.003) | 1 | W |
| Enables load control | | | | |
| **DPTx_TimedDynamicMode** | **DP_DynamicMode** | DPT_Start (1.010) | 2 | W |
| Starts or stops the Dynamic Mode for the DP_ValueF, for greater visualization accuracy of the data acquired from the input channel. | | | | |
| **DPTx_HumiditySetpoint *** | **DP_Setpoint** | DPT_Value_Temp(9.0xx) | 3 | W |
| Changes the set-point value | | | | |
| **DPTx_HumidityHysteresis *** | **DP_Hyst** | DPT_Value_Temp(9.0xx) | 4 | W |
| Changes the hysteresis value | | | | |
| **DPTx_HumidityValue *** | **DP_ValueF** | DPT_Value_Temp(9.0xx) | 5 | RT |
| Value converted in the range PAR_ValueAtMin and PAR_ValueAtMax of the input channel. | | | | |
| **DPTx_Value** | **DP_ValueS** | DPT_Percent_U8(5,004) | 6 | RT |
| Percentage value converted in the range PAR_ValueAtMin and PAR_ValueAtMax of the input channel | | | | |
| **DPTx_OnOff** | **DP_SwitchOnOff** | DPT_Switch(1.001) | 7 | RT |
| Load on / off command. | | | | |
| **DPTx_EnableInfo** | **DP_InfoCtrlEnable** | DPT_Enable(1.003) | 8 | RT |
| Info of the DP_Enable. | | | | |
| **DPTx_HumiditySetpointInfo *** | **DP_InfoSetpoint** | DPT_Value_Temp(9.0xx) | 9 | RT |
| Info of the DP_SetPoint | | | | |
| **DPTx_HumidityHysteresisInfo *** | **DP_InfoHyst** | DPT_Value_Temp(9.0xx) | 10 | RT |
| Info of the DP_Hyst | | | | |

### 3.4.2 FBT_3IN_VISUAL_TEMP

#### 3.4.2.1 Functional block




#### 3.4.2.2 Description of Communication Objects

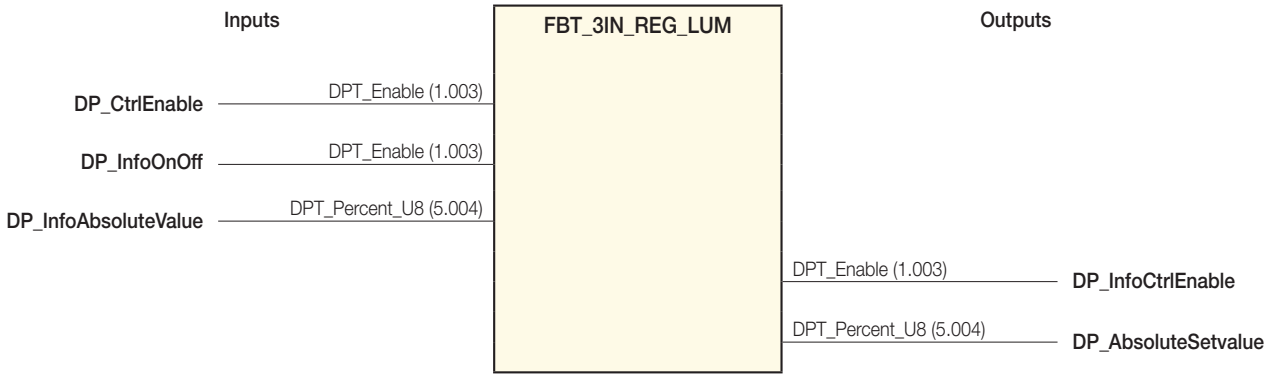
The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_TemperatureValue** | **DP_Temperature** | DPT_Value_Temp(9.001) | 1 | RT |
| Temperature read from the NTC input | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.4.3 FBT_3IN_REG_LUM

#### 3.4.3.1 Functional block




#### 3.4.3.2 Description of Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_Enable** | **DP_CtrlEnable** | DPT_Enable(1.003) | 1 | W |
| Enables automatic ambient brightness control | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Enable(1.003) | 2 | W |
| Suspends ambient brightness control | | | | |
| **DPTx_BrightnessInfo** | **DP_InfoAbsoluteValue** | DPT_Percent_U8 (5.004) | 3 | W |
| Dimmer control setpoint | | | | |
| **DPTx_EnableInfo** | **DP_InfoCtrlEnable** | DPT_Enable(1.003) | 5 | RT |
| Info on enabling automatic ambient brightness control | | | | |
| **DPTx_Brightness** | **DP_AbsoluteSetvalue** | DPT_Percent_U8 (5.004) | 6 | RT |
| Dimmer output to control the dimmer according to the control algorithm (to maintain the required setpoint) | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.5 Eikon Tactil FAN thermostat (21514.F)

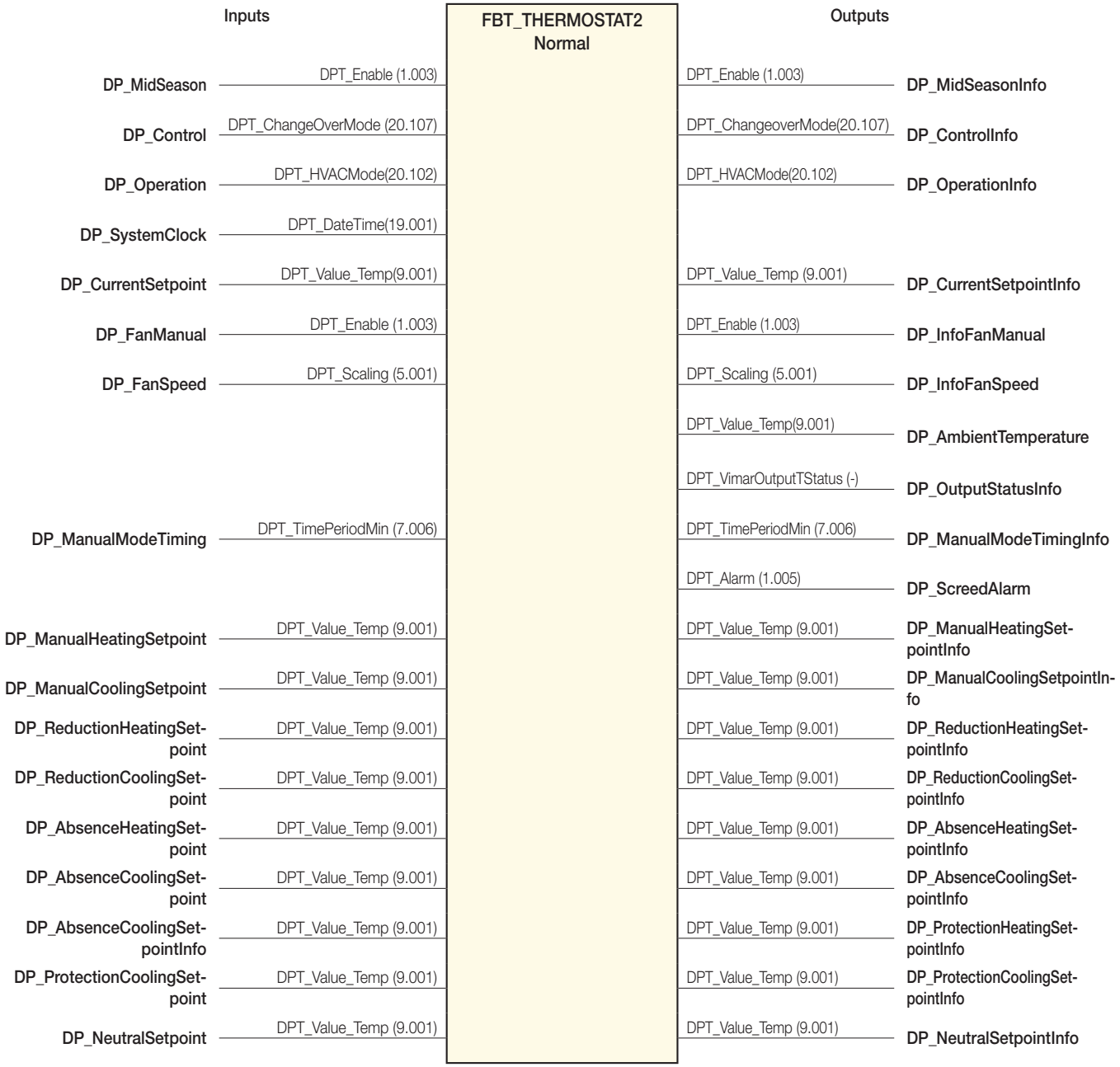
The device consists of 10 Functional Blocks.

| Fb Index | Fb Name | Fb Type |
|---|---|---|
| **1** | FB_Thermostat | FBT_THERMOSTAT2 (0x7A) |
| Thermostat | | |
| **2** | FB_Output1 | FBT_OUTPUT (0x5C) |
| Output 1 control | | |
| **3** | FB_Output2 | FBT_OUTPUT (0x5C) |
| Output 2 control | | |
| **4** | FB_Output3 | FBT_OUTPUT (0x5C) |
| Output 3 control | | |
| **5** | FB_Output4 | FBT_OUTPUT (0x5C) |
| Output 4 control | | |
| **6** | FB_Probe2Temp | FBT_THERM_PROBE2 (0x7B) |
| Probe 2 temperature | | |
| **7** | FB_Alarm | FBT_THERM_ALARM_TX (0x7C) |
| Window open/boiler alarm from device terminal | | |
| **8** | FB_AlarmWindow | FBT_THERM_WINDOW_RX (0x7D) |
| Window open alarm from BUS | | |
| **9** | FB_AlarmBlock | FBT_THERM_BLOCK_RX (0x7E) |
| Boiler shutdown alarm from BUS | | |
| **10** | FB_Humidity | FBT_REG_HUMIDITY (0x5F) |
| Humidity value | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.5.1 FBT_THERMOSTAT2

#### 3.5.1.1 “Normal” Channel



## SECTION 4: TEMPERATURE CONTROL

### 3.5.1.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_MidSeasonEnable** | **DP_MidSeason** | DPT_Enable (1.003) | 1 | W |
| Mid-season function enable control (main output disabled and auxiliary output enabled). | | | | |
| **DPTx_MidSeasonEnableInfo** | **DP_MidSeasonInfo** | DPT_Enable (1.003) | 2 | RT |
| Provides the current status of the mid-season function. | | | | |
| **DPTx_ChangeOverMode** | **DP_Control** | DPT_ChangeOverMode (20.107) | 3 | W |
| Command for changing control mode:<br>0 = neutral zone (auto)<br>1 = air-conditioning (cooling only)<br>2 = heating (heating only)<br>* in integration mode (PAR_AdjustableSelect=Integration)<br>* in neutral zone, it is used to limit the operating mode<br>* on actually changing the operating mode, a message is generated by DP_ControlInfo | | | | |
| **DPTx_ChangeOverModeInfo** | **DP_ControlInfo** | DPT_ChangeoverMode(20.107) | 4 | RT |
| Provides the current status of the control mode.<br>It is sent on changing the mode. | | | | |
| **DPTx_HvacMode** | **DP_Operation** | DPT_HVACMode(20.102) | 5 | W |
| Command for changing operating mode:<br>KNX - VIMAR<br>(0) auto = auto<br>(1) comfort = manual<br>(2) standby = reduction<br>(3) economy = absence<br>(4) building protection = protection (anti-freeze/too hot)<br>(5) / = manual, with timer<br>(6) / = off<br>* Changes the operating mode subject to the limitations on use and constraints of changing mode<br>* In integration mode, it is used to view the operating modes of the thermostat on the App and on the touch screens, to receive commands from a viewer and forward them to the third-party system via the corresponding Info DP. | | | | |
| **DPTx_HvacModeInfo** | **DP_OperationInfo** | DPT_HVACMode(20.102) | 6 | RT |
| Provides the current status of the operating mode.<br>It is sent on changing the mode.<br>* In integration it is used to control the third-party system in the operating mode selected on the thermostat or received from the viewer | | | | |
| **DPTx_SysClockDateTime** | **DP_SystemClock** | DPT_DateTime(19.001) | 7 | W |
| Updates the device clock.<br>Used for the time schedule.<br>When the device is started it asks the system for the time, if this does not arrive it works with its own internal clock. | | | | |
| **DPTx_TemperatureSetpoint1** | **DP_CurrentSetpoint** | DPT_Value_Temp(9.001) | 8 | W |
| Used to set the current setpoint value. This value is normally propagated to the relative DP_Setpoint* of the current operating mode, in some cases this does not occur. Read the specifications. | | | | |
| **DPTx_TemperatureSetpointInfo1** | **DP_CurrentSetpointInfo** | DPT_Value_Temp(9.001) | 9 | RT |
| Provides the current setpoint value.<br>It is sent upon a change.<br>So it is also sent when the control mode or operating mode change. | | | | |
| **DPTx_FanManualEnable** | **DP_FanManual** | DPT_Enable(1.003) | 10 | W |
| Indicates the type of fan coil fan control:<br>0 = automatic<br>1 = manual<br>in the case of manual speed, it uses the value in DP_FanSpeed to control the speed of the fans. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_FanManualEnableInfo** | **DP_InfoFanManual** | DPT_Enable(1.003) | 11 | RT |
| Provides the status of the relevant DP.<br>It is sent upon a change. | | | | |
| **DPTx_FanSpeed** | **DP_FanSpeed** | DPT_Scaling(5.001) | 12 | W |
| For the manual control of the fan coil fans, it indicates the value to control the fan speed at.<br>In the case of a 3-speed fan coil:<br>[0%,33%] => V1<br>[33%, 67%] => V2<br>[67%, 100%] => V3<br>CAUTION: 0% (fans off, OFF) switching off the fans is not allowed<br><br>On the contrary, in integration mode the value of 0% may change the graphic symbol to OFF, indicating that the fans have been switched off by the third-party system. | | | | |
| **DPTx_FanSpeedInfo** | **DP_InfoFanSpeed** | DPT_Scaling(5.001) | 13 | RT |
| Provides the current status of the fan speed.<br>It is sent upon a change.<br>If there is no fan coil as an output device or the fans are off it returns to 0%.<br><br>In the case of a 3-speed fan coil, the DP admits the following values:<br>V1 = 33% (0x55)<br>V2 = 67% (0xAA)<br>V3 = 100% (0xFF) | | | | |
| **DPTx_AmbientTemperature** | **DP_AmbientTemperature** | DPT_Value_Temp(9.001) | 30 | RT |
| Provides the temperature value used by the device to perform temperature control.<br>This value can be composed of:<br>- Probe 1<br>- Probe 2<br>- Average between Probe 1 and Probe 2<br><br>It is sent with a variation of 0.1°C<br><br>In integration mode it serves to provide third parties with the temperature value measured by the probe. | | | | |
| **DPTx_OutStatus** | **DP_OutputStatusInfo** | DPT_VimarOutputTStatus(-) | 31 | RT |
| Provides the current status of all the thermostat outputs. Only on/off status.<br>It is used for supervisors, to have summary information without checking all the outputs.<br><br>Basically it is a representation of the active output icon on the thermostat GUI.<br><br>It is sent upon a change.<br><br>The format:<br><br>DPT_State[3] (1.011)<br><br>Format: 1 octet: B3<br>octet no: 1<br><br>0 = off<br>1 = on | | | | |

| | | | | | Main Heat. | Main Cool. | Boost |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

| DPTx_ManualModeTiming | DP_ManualModeTiming | DPT_TimePeriodMin(7.006) | 37 | W |
|---|---|---|---|---|
| Value used by the thermostat during the "timed manual" operating mode.<br>When changing the mode to "timed manual", the timer is started with a value equal to this DP.<br>During "timed manual" mode, a change in this DP restarts the timer on the new value.<br>At the end of the set time, the operating mode returns to the previous one.<br><br>Value not permitted: 0<br><br>Example:<br>By default it starts at 1h, then an update will come from the DP which will restart the timer and restart the count. The next time it will start with the last value received. | | | | |

## SECTION 4: TEMPERATURE CONTROL

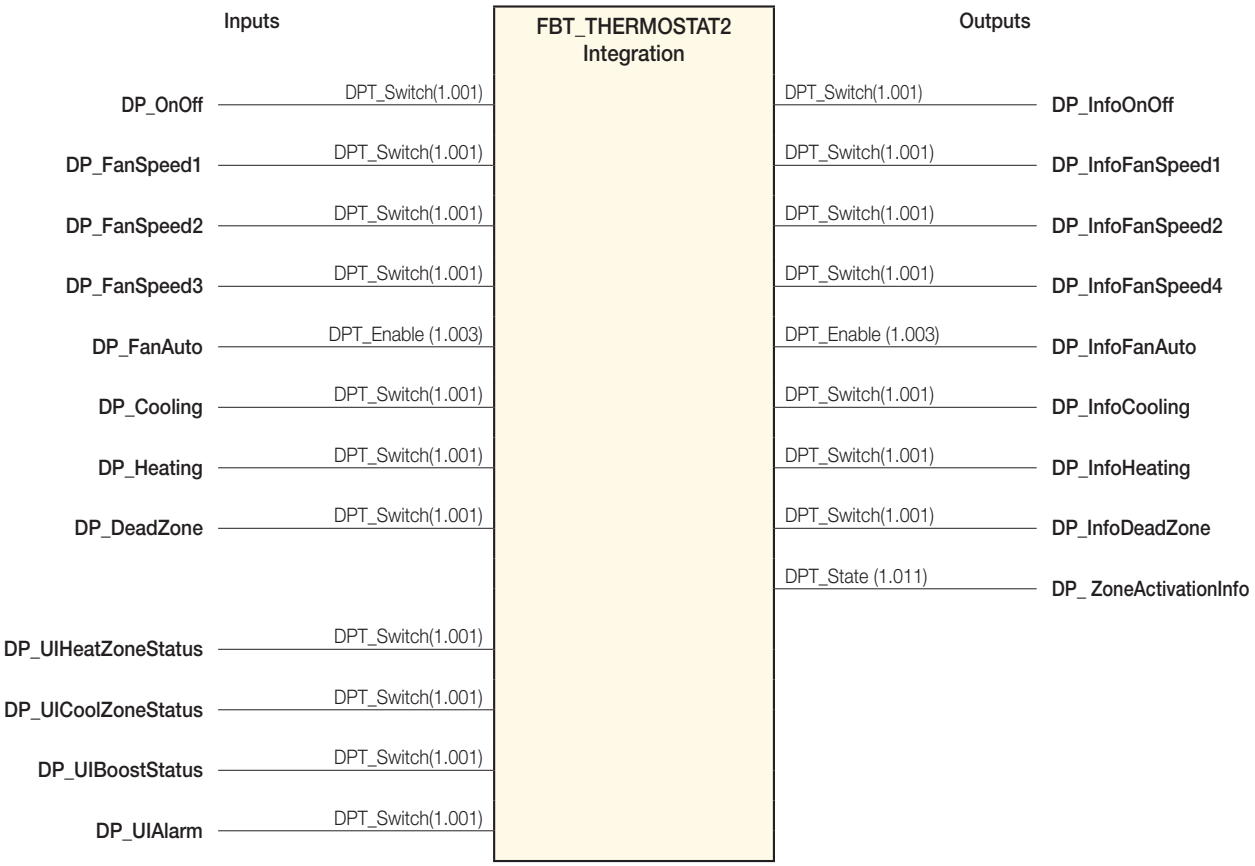
| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ManualModeTimingInfo** | **DP_ManualModeTimingInfo** | DPT_TimePeriodMin(7.006) | 38 | RT |
| Provides the value of the relevant DP.<br>It is sent upon a change. | | | | |
| **DPTx_SubFloorAlarm** | **DP_ScreedAlarm** | DPT_Alarm(1.005) | 39 | RT |
| It is generated in the event of a screed limitation alarm.<br>It remains set on 1 until the alarm ends (interaction by the user is needed to reset the alarm) | | | | |
| **DPTx_ManualHeatingSetpoint** | **DP_ManualHeatingSetpoint** | DPT_Value_Temp(9.001) | 40 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ManualHeatingSetpointInfo** | **DP_ManualHeatingSetpointInfo** | DPT_Value_Temp(9.001) | 41 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_ManualCoolingSetpoint** | **DP_ManualCoolingSetpoint** | DPT_Value_Temp(9.001) | 42 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ManualCoolingSetpointInfo** | **DP_ManualCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 43 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_ReductionHeatingSetpoint** | **DP_ReductionHeatingSetpoint** | DPT_Value_Temp(9.001) | 44 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ReductionHeatingSetpointInfo** | **DP_ReductionHeatingSetpointInfo** | DPT_Value_Temp(9.001) | 45 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_ReductionCoolingSetpoint** | **DP_ReductionCoolingSetpoint** | DPT_Value_Temp(9.001) | 46 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ReductionCoolingSetpointInfo** | **DP_ReductionCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 47 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_AbsenceHeatingSetpoint** | **DP_AbsenceHeatingSetpoint** | DPT_Value_Temp(9.001) | 48 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_AbsenceHeatingSetpointInfo** | **DP_AbsenceHeatingSetpointInfo** | DPT_Value_Temp(9.001) | 49 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_AbsenceCoolingSetpoint** | **DP_AbsenceCoolingSetpoint** | DPT_Value_Temp(9.001) | 50 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_AbsenceCoolingSetpointInfo** | **DP_AbsenceCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 51 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ProtectionHeatingSetpoint** | **DP_ProtectionCoolingSetpoint** | DPT_Value_Temp(9.001) | 52 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ProtectionHeatingSet-pointInfo** | **DP_ProtectionHeatingSetpointInfo** | **DPT_Value_Temp(9.001)** | 53 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_ProtectionCoolingSetpoint** | **DP_ProtectionCoolingSetpoint** | DPT_Value_Temp(9.001) | 54 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_ProtectionCoolingSet-pointInfo** | **DP_ProtectionCoolingSetpointInfo** | DPT_Value_Temp(9.001) | 55 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |
| **DPTx_NeutralSetpoint** | **DP_NeutralSetpoint** | DPT_Value_Temp(9.001) | 56 | W |
| Direct access to the setpoint relating to the operation and control mode indicated by the DP name. | | | | |
| **DPTx_NeutralSetpointInfo** | **DP_NeutralSetpointInfo** | DPT_Value_Temp(9.001) | 57 | RT |
| Setpoint value relating to the operation and control mode indicated by the DP name.<br>Sent on a change in setpoint, from whatever source it comes from (DP_CurrentSetpoint or DP_Setpoint*) | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.5.1.3 “Integration” Channel




### 3.5.1.4 Description of the Communication Objects

The "Integration" channel provides for the following Communication Objects, *in addition to those already provided for by the "normal" channel.*

| Name in View Pro | Datapoint | DPT | CO num-ber | Flags |
|---|---|---|---|---|
| **DPTx_ThermostatOnOff** | **DP_OnOff** | DPT_Switch(1.001) | 14 | W |
| Indicates the status the thermostat should be in:<br>0 = Operating Mode OFF<br>1 = Operating Mode other than OFF | | | | |
| **DPTx_InfoThermostatOnOff** | **DP_InfoOnOff** | DPT_Switch(1.001) | 15 | RT |
| Provides the current status of the operating mode:<br>0 = Operating Mode OFF<br>1 = Operating Mode other than OFF | | | | |
| **DPTx_FanSpeed1** | **DP_FanSpeed1** | DPT_Switch(1.001) | 16 | W |
| Sets the speed V1 of the fan coil fan. | | | | |
| **DPTx_InfoFanSpeed1** | **DP_InfoFanSpeed1** | DPT_Switch(1.001) | 17 | RT |
| Indicates the speed V1 of the fan coil fan. | | | | |
| **DPTx_FanSpeed2** | **DP_FanSpeed2** | DPT_Switch(1.001) | 18 | W |
| Sets the speed V2 of the fan coil fan. | | | | |
| **DPTx_InfoFanSpeed2** | **DP_InfoFanSpeed2** | DPT_Switch(1.001) | 19 | RT |
| Indicates the speed V2 of the fan coil fan. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_FanSpeed3** | **DP_FanSpeed3** | DPT_Switch(1.001) | 20 | W |
| Sets the speed V3 of the fan coil fan. | | | | |
| **DPTx_InfoFanSpeed3** | **DP_InfoFanSpeed3** | DPT_Switch(1.001) | 21 | RT |
| Indicates the speed V3 of the fan coil fan. | | | | |
| **DPTx_FanAuto** | **DP_FanAuto** | DPT_Enable(1.003) | 22 | W |
| Indicates the type of fan coil fan control:<br>0 = manual<br>1 = automatic<br><br>It is used in the event of a splitter integration where the semantics are inverted compared to the DP_FanManual. | | | | |
| **DPTx_InfoFanAuto** | **DP_InfoFanAuto** | DPT_Enable(1.003) | 23 | RT |
| Provides the status of the relevant DP. | | | | |
| **DPTx_Cooling** | **DPT_Cooling** | DPT_Switch(1.001) | 24 | W |
| Sets the status of the Air Conditioning control mode<br>0 = off<br>1 = on | | | | |
| **DPTx_InfoCooling** | **DP_InfoCooling** | DPT_Switch(1.001) | 25 | RT |
| Indicates the status of the Air Conditioning control mode | | | | |
| **DPTx_Heating** | **DPT_Heating** | DPT_Switch(1.001) | 26 | W |
| Sets the status of the Heating control mode<br>0 = off<br>1 = on | | | | |
| **DPTx_InfoHeating** | **DPT_InfoHeating** | DPT_Switch(1.001) | 27 | RT |
| Indicates the status of the Heating control mode | | | | |
| **DPTx_DeadZone** | **DPT_DeadZone** | DPT_Switch(1.001) | 28 | W |
| Sets the status of the Neutral Zone control mode<br>0 = off<br>1 = on | | | | |
| **DPTx_InfoDeadZone** | **DPT_InfoDeadZone** | DPT_Switch(1.001) | 29 | RT |
| Indicates the status of the Neutral Zone control mode | | | | |
| **DPTx_ZoneActivationInfo** | **DPT_ZoneActivationInfo** | DPT_State (1.011) | 32 | RT |
| Provides the status of the zone. That is, it informs if the thermostat is on and not off.<br>It is used for third-party integration to understand, with a single DP, when the system must be switched on or off.<br>For example, this DP is NOT used by the Circulation Pump device to understand when to switch on, because it provides no information on the status of the Heat/Cool output.<br><br>It is sent upon a change.<br>0 = zone off<br>1 = if DP_OperationInfo!= 6 (OFF) | | | | |
| **DPTx_HeatingStatus** | **DP_UIHeatZoneStatus** | DPT_Switch(1.001) | 33 | W |
| Used in integration to manage the “active output” icon. | | | | |
| **DPTx_CoolingStatus** | **DP_UICoolZoneStatus** | DPT_Switch(1.001) | 34 | W |
| Used in integration to manage the “active output” icon. | | | | |
| **DPTx_BoostStatus** | **DP_UIBoostStatus** | DPT_Switch(1.001) | 35 | W |
| Used in integration to manage the boost status. | | | | |
| **DPTx_UIAlarm** | **DP_UIAlarm** | DPT_Switch(1.001) | 36 | W |
| Used in integration to manage the alarm status (flashing display) | | | | |

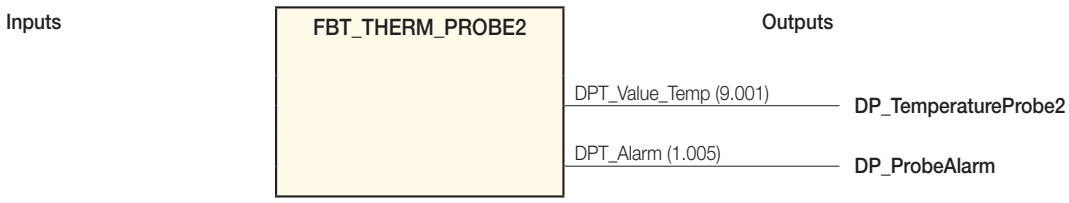
## SECTION 4: TEMPERATURE CONTROL

### 3.5.2 FBT_OUTPUT

For the description of the communication datapoints please refer to the "3.1.2 FBT_OUTPUT" chapter since the functional blocks have the same model.

### 3.5.3 FBT_THERM_PROBE2

#### 3.5.3.1 Functional Block



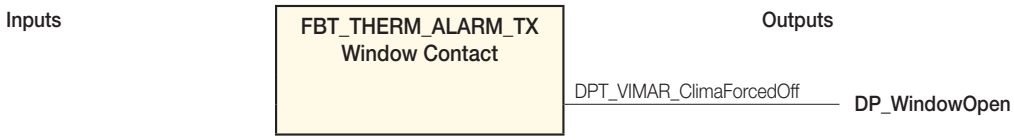

#### 3.5.3.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_TemperatureValue** | **DP_TemperatureProbe2** | DPT_Value_Temp (9.001) | 1 | RT |
| Temperature read by the probe connected to the device terminals. | | | | |
| **DPTx_Alarm** | **DP_ProbeAlarm** | DPT_Switch(1.001) | 2 | RT |
| Alarm signal. Sent in the case of a status change. Sent in the case of probe failure (short or open). | | | | |

### 3.5.4 FBT_THERM_ALARM_TX

#### 3.5.4.1 "Window Contact" Channel

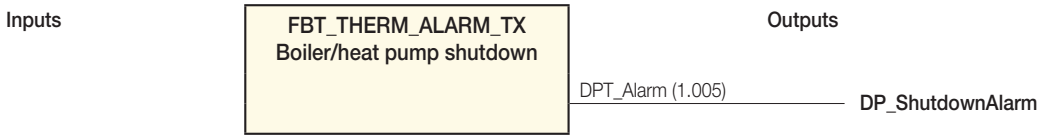



#### 3.5.4.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_ContactClosed** | **DP_WindowOpen** | DPT_VIMAR_ClimaForcedOff | 1 | RT |
| Indicates the window open condition, detected by the on-board terminal.<br>Format:<br>0 = open window (request for forcing the system OFF)<br>1 = window closed | | | | |

#### 3.5.4.3 "Boiler/heat pump shutdown" Channel



## SECTION 4: TEMPERATURE CONTROL

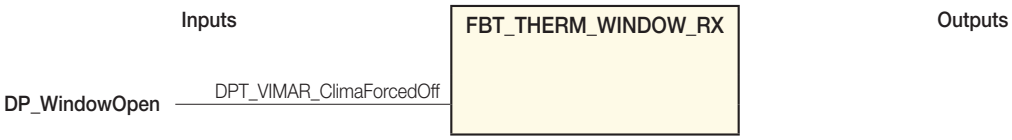
#### 3.5.4.4 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| DPTx_ContactClosed | DP_ShutdownAlarm | DPT_Alarm (1.005) | 2 | RT |
| Indicates the boiler/heat pump shutdown condition, detected by the on-board terminal. | | | | |

### 3.5.5 FBT_THERM_WINDOW_RX

#### 3.5.5.1 Functional Block



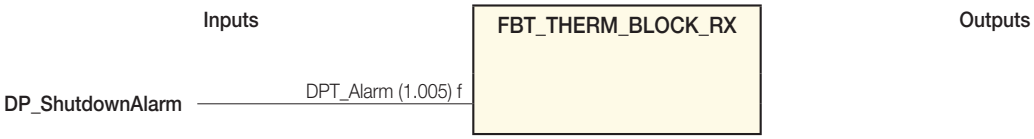

#### 3.5.5.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| DPTx_ContactClosed | DP_WindowOpen | DPT_VIMAR_ClimaForcedOff | 1 | W |
| Receives the window open signal from the bus.<br>Format:<br>0 = open window (request for forcing the system OFF)<br>1 = window closed | | | | |

### 3.5.6 FBT_THERM_BLOCK_RX

#### 3.5.6.1 Functional Block




#### 3.5.6.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| DPTx_ContactClosed (*) | DP_ShutdownAlarm | DPT_Alarm (1.005) | 1 | W |
| Receives the boiler shutdown signal from the bus. | | | | |

### 3.5.7 FBT_HUMIDITY

For the description of the communication datapoints please refer to the “3.2.3 FBT_” REG_HUMIDITY” chapter since the functional blocks have the same model.

# SECTION 4: TEMPERATURE CONTROL

### 3.6 Eikon Tactil HOTEL thermostat (21514.H)

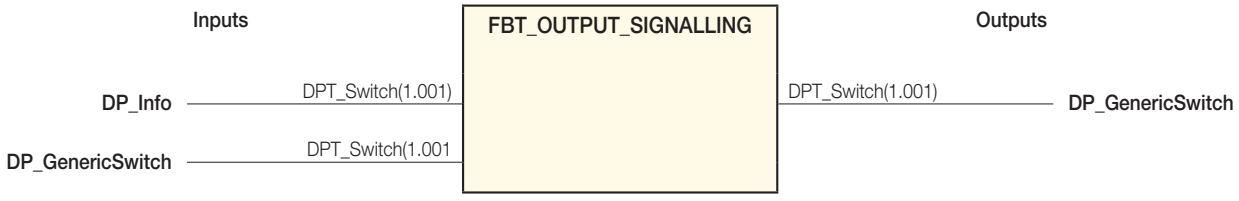
The device consists of 11 Functional Blocks.

| Fb Index | Fb Name | Fb Type |
|---|---|---|
| **1** | FB_Thermostat | FBT_THERMOSTAT2 (0x7A) |
| Thermostat | | |
| **2** | FB_Output1 | FBT_OUTPUT (0x5C) |
| Output 1 control | | |
| **3** | FB_Output2 | FBT_OUTPUT (0x5C) |
| Output 2 control | | |
| **4** | FB_Output3 | FBT_OUTPUT (0x5C) |
| Output 3 control | | |
| **5** | FB_Output4 | FBT_OUTPUT (0x5C) |
| Output 4 control | | |
| **6** | FB_Probe2Temp | FBT_THERM_PROBE2 (0x7B) |
| Probe 2 temperature | | |
| **7** | FB_Alarm | FBT_THERM_ALARM_TX (0x7C) |
| Window open/boiler alarm from device terminal | | |
| **8** | FB_AlarmWindow | FBT_THERM_WINDOW_RX (0x7D) |
| Window open alarm from BUS | | |
| **9** | FB_AlarmBlock | FBT_THERM_BLOCK_RX (0x7E) |
| Boiler shutdown alarm from BUS | | |
| **10** | FB_Dnd | FBT_OUTPUT_SIGNALLING (0x80) |
| "DO NOT DISTURB" signalling | | |
| **11** | **FB_Mkr** | FBT_OUTPUT_SIGNALLING (0x80) |
| "MAKE UP ROOM" signalling | | |

For the description of the functional blocks not listed in the following paragraphs, please refer to paragraph 3.5 as they are similar.

#### 3.6.1 FBT_OUTPUT_SIGNALLING

##### 3.6.1.1 Functional Block




##### 3.6.1.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo** | **DP_Info** | DPT_Switch (1.001) | 1 | W |
| Feedback of the associated actuator status. | | | | |
| **DPTx_OnOff** | **DP_GenericSwitch** | DPT_Switch(1.001) | 2 | RWT |
| Feedback of the associated actuator status. | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 3.7 Eikon Tactil STAR thermostat (21514.S)

The device consists of 11 Functional Blocks.

| Fb Index | Fb Name | Fb Type |
|---|---|---|
| **1** | FB_Thermostat | FBT_THERMOSTAT2 (0x7A) |
| Thermostat | | |
| **2** | FB_Output1 | FBT_OUTPUT (0x5C) |
| Output 1 control | | |
| **3** | FB_Output2 | FBT_OUTPUT (0x5C) |
| Output 2 control | | |
| **4** | FB_Output3 | FBT_OUTPUT (0x5C) |
| Output 3 control | | |
| **5** | FB_Output4 | FBT_OUTPUT (0x5C) |
| Output 4 control | | |
| **6** | FB_Probe2Temp | FBT_THERM_PROBE2 (0x7B) |
| Probe 2 temperature | | |
| **7** | FB_Alarm | FBT_THERM_ALARM_TX (0x7C) |
| Window open/boiler alarm from device terminal | | |
| **8** | FB_AlarmWindow | FBT_THERM_WINDOW_RX (0x7D) |
| Window open alarm from BUS | | |
| **9** | FB_AlarmBlock | FBT_THERM_BLOCK_RX (0x7E) |
| Boiler shutdown alarm from BUS | | |
| **10** | FB_StarKey | FBT_BUTTON_TOUCH (0x71) |
| Function of key ☆ | | |
| **11** | **FB_Humidity** | FBT_REG_HUMIDITY (0x5F) |
| Humidity value | | |

For the description of the functional blocks not listed in the following paragraphs, please refer to paragraph 3.5 as they are similar.

#### 3.7.1 FBT_BUTTON_TOUCH

For the description of the communication datapoints please refer to the 4.1 "Functional block “VIMAR 3.0 Single push buttons” (FBT_BUTTON)" paragraph in "SECTION 2: LIGHT AND ROLLER SHUTTER CONTROL" since the functional blocks have the same model.

## SECTION 4: TEMPERATURE CONTROL

### 3.8 Home automation system dial thermostat (02971)

The device consists of 10 Functional Blocks.

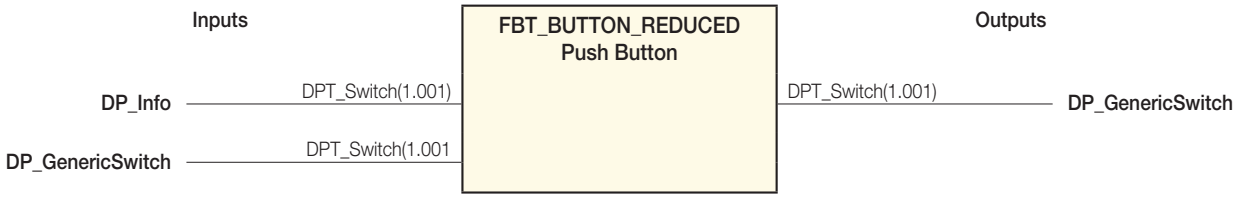
| Fb Index | Fb Name | Fb Type |
|---|---|---|
| 1 | FB_Thermostat | FBT_THERMOSTAT2 (0x7A) |
| Thermostat | | |
| 2 | FB_Output1 | FBT_OUTPUT (0x5C) |
| Output 1 control | | |
| 3 | FB_Output2 | FBT_OUTPUT (0x5C) |
| Output 2 control | | |
| 4 | FB_Output3 | FBT_OUTPUT (0x5C) |
| Output 3 control | | |
| 5 | FB_Output4 | FBT_OUTPUT (0x5C) |
| Output 4 control | | |
| 6 | FB_Probe2Temp | FBT_THERM_PROBE2 (0x7B) |
| Probe 2 temperature | | |
| 7 | FB_Alarm | FBT_THERM_ALARM_TX (0x7C) |
| Window open/boiler alarm from device terminal | | |
| 8 | FB_AlarmWindow | FBT_THERM_WINDOW_RX (0x7D) |
| Window open alarm from BUS | | |
| 9 | FB_AlarmBlock | FBT_THERM_BLOCK_RX (0x7E) |
| Boiler shutdown alarm from BUS | | |
| 10 | FB_StarKey | FBT_BUTTON_REDUCED (0x7F) |
| Function of key ☆ | | |

For the description of the functional blocks not listed in the following paragraphs, please refer to paragraph 3.5 as they are similar.

## SECTION 4: TEMPERATURE CONTROL

### 3.8.1 FBT_BUTTON_REDUCED

#### 3.8.1.1 “Push button” Channel

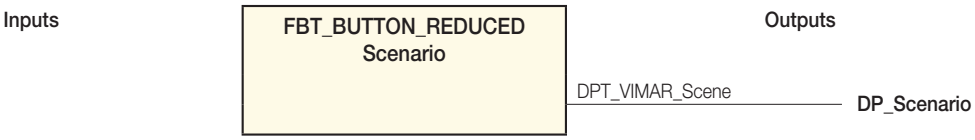



#### 3.8.1.2 Description of the Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOffInfo (*)** | **DP_Info** | DPT_Switch (1.001) | 1 | W |
| Feedback of the associated actuator status. | | | | |
| **DPTx_OnOff (*)** | **DP_GenericSwitch** | DPT_Switch(1.001) | 2 | RWT |
| Feedback of the associated actuator status. | | | | |

#### 3.8.1.3 “Scenario” Channel




#### 3.8.1.4 Description of the Communication Objects

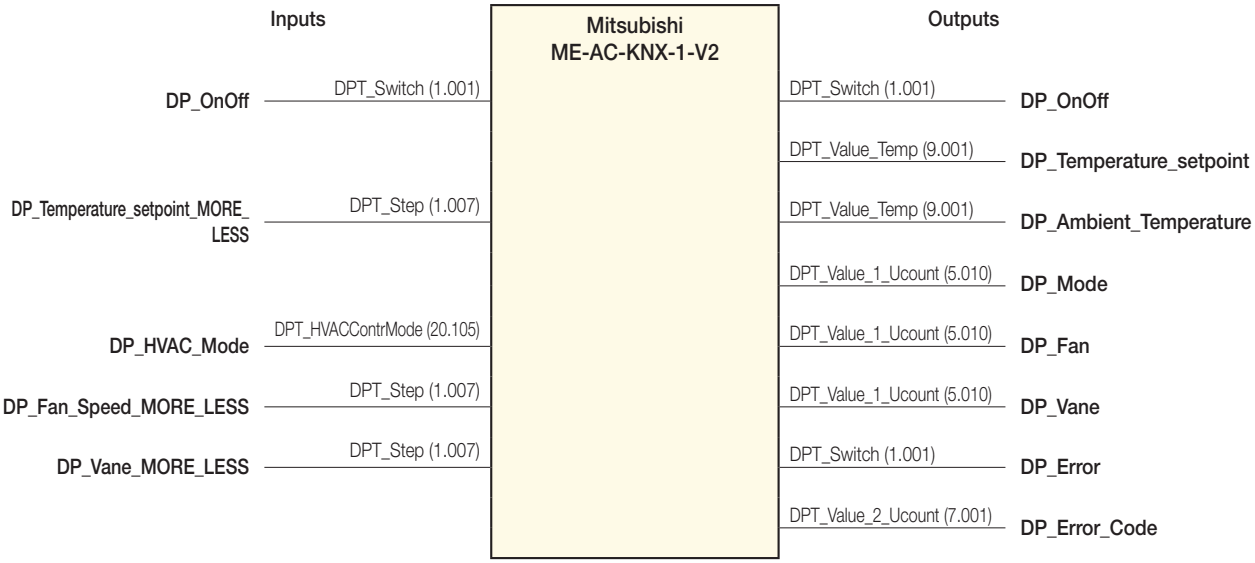
The detailed descriptions of each Datapoint covered by the model are given below.

| Name in View Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_SceneActivator1** | **DP_Scenario** | DPT_VIMAR_Scene | 3 | T |
| Control for calling up a scenario. | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 4. KNX interfacing

#### 4.1 Mitsubishi via ME-AC-KNX-1-V2 with databank version 0.8

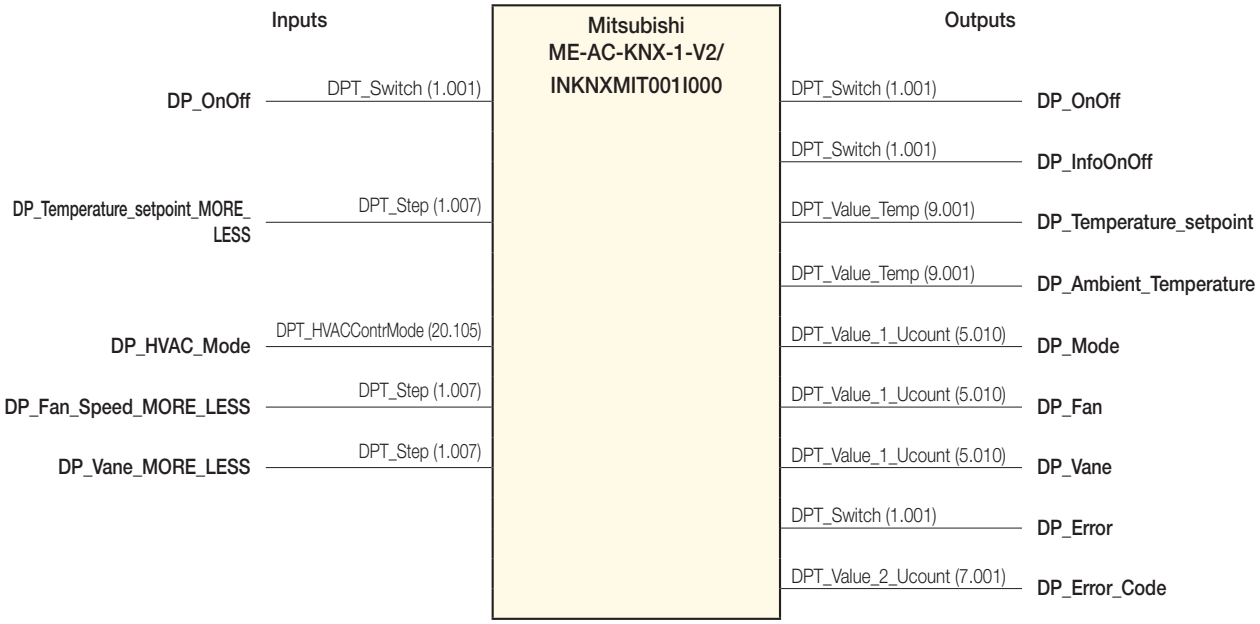



| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_OnOff** | DPT_Switch (1.001) | 0 | RWT |
| Split control and on status. | | | | |
| **DPTx_TemperatureSetpoint1** | **DP_Temperature_setpoint** | DPT_Value_Temp (9.001) | 7 | RT |
| Currently set setpoint value. | | | | |
| **DPTx_TemperatureSetpointStep** | **DP_Temperature_setpoint_MORE_LESS** | DPT_Step (1.007) | 38 | W |
| Setpoint up/down control. | | | | |
| **DPTx_AmbientTemperature** | **DP_Ambient_Temperature** | DPT_Value_Temp (9.001) | 8 | RT |
| Ambient temperature value measured by the split. | | | | |
| **DPTx_HVACContrMode *** | **DP_HVAC_Mode** | DPT_HVACContrMode (20.105) | 49 | W |
| Command for setting the operating mode. | | | | |
| **DPTx_HVACContrModeInfo** | **DP_Mode** | DPT_Value_1_Ucount (5.010) | 1 | RT |
| Currently set operating mode status. | | | | |
| **DPTx_FanSpeedStep** | **DP_Fan_Speed_MORE_LESS** | DPT_Step (1.007) | 24 | W |
| Fan speed up/down control. | | | | |
| **DPTx_InfoFan** | **DP_Fan** | DPT_Value_1_Ucount (5.010) | 3 | RT |
| Fan speed status. | | | | |
| **DPTx_VaneStep** | **DP_Vane_MORE_LESS** | DPT_Step (1.007) | 30 | W |
| Slat position up/down control. | | | | |
| **DPTx_InfoVane** | **DP_Vane** | DPT_Value_1_Ucount (5.010) | 5 | RT |
| Slat position status. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_Error** | **DP_Error** | DPT_Switch (1.001) | 9 | RT |
| Split error condition status. | | | | |
| **DPTx_ErrorCode** | **DP_Error_Code** | DPT_Value_2_Ucount (7.001) | 10 | RT |
| Error code value. | | | | |

### 4.2 Mitsubishi via ME-AC-KNX-1-V2/INKNXMIT001I000 with databank version 1.x

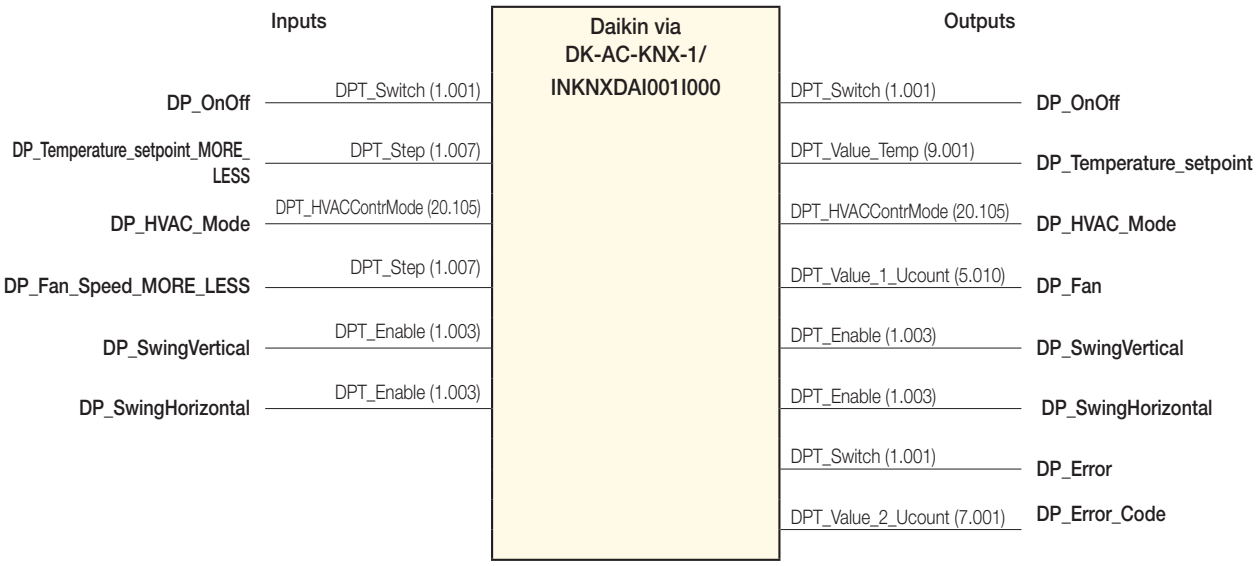



| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_OnOff** | DPT_Switch (1.001) | 0 | WT |
| Split on control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 43 | RT |
| Split on status. | | | | |
| **DPTx_TemperatureSetpointInfo1** | **DP_Temperature_setpoint** | DPT_Value_Temp (9.001) | 68 | RT |
| Currently set setpoint value. | | | | |
| **DPTx_TemperatureSetpointStep** | **DP_Temperature_setpoint_MORE_LESS** | DPT_Step (1.007) | 26 | W |
| Setpoint up/down control. | | | | |
| **DPTx_AmbientTemperature** | **DP_Ambient_Temperature** | DPT_Value_Temp (9.001) | 69 | RT |
| Ambient temperature value measured by the split. | | | | |
| **DPTx_HVACContrMode *** | **DP_HVAC_Mode** | DPT_HVACContrMode (20.105) | 1 | W |
| Command for setting the operating mode. | | | | |
| **DPTx_HVACContrModeInfo** | **DP_Mode** | DPT_Value_1_Ucount (5.010) | 76 | RT |
| Currently set operating mode status. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_FanSpeedStep** | **DP_Fan_Speed_MORE_LESS** | DPT_Step (1.007) | 15 | W |
| Fan speed up/down control. | | | | |
| **DPTx_InfoFan** | **DP_Fan** | DPT_Value_1_Ucount (5.010) | 77 | RT |
| Fan speed status. | | | | |
| **DPTx_VaneStep** | **DP_Vane_MORE_LESS** | DPT_Step (1.007) | 24 | W |
| Slat position up/down control. | | | | |
| **DPTx_InfoVane** | **DP_Vane** | DPT_Value_1_Ucount (5.010) | 78 | RT |
| Slat position status. | | | | |
| **DPTx_Error** | **DP_Error** | DPT_Switch (1.001) | 70 | RT |
| Split error condition status. | | | | |
| **DPTx_ErrorCode** | **DP_Error_Code** | DPT_Value_2_Ucount (7.001) | 71 | RT |
| Error code value. | | | | |

### 4.3 Daikin via DK-AC-KNX-1/INKNXDAI001I000 with databank version 0.4

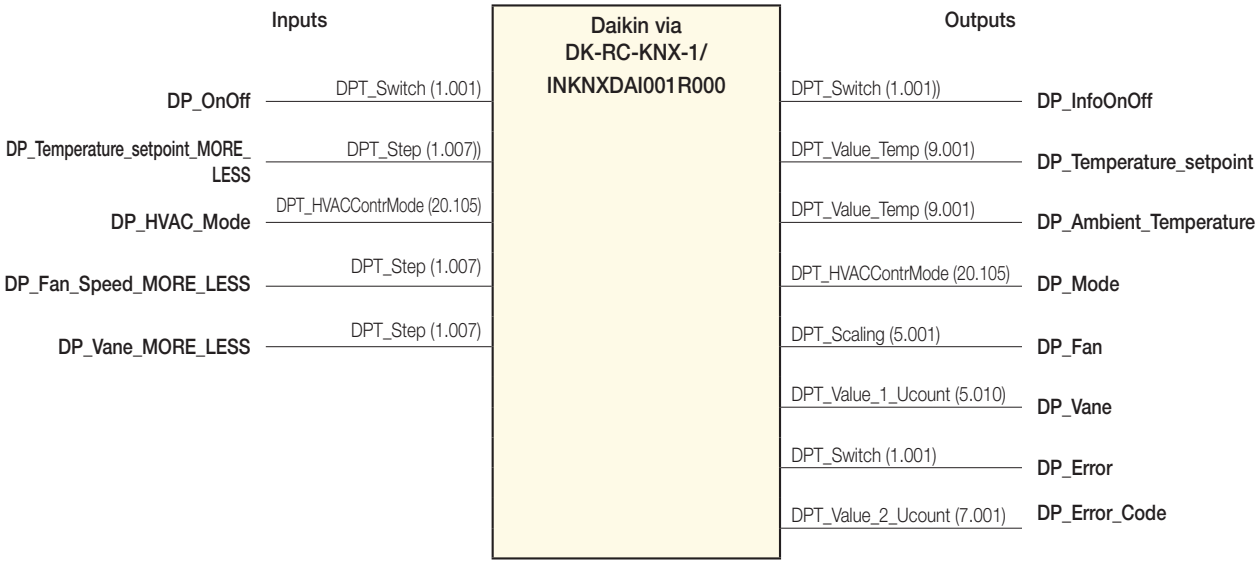



| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_OnOff** | DPT_Switch (1.001) | 0 | RWT |
| Split on control. | | | | |
| **DPTx_TemperatureSetpointInfo1** | **DP_Temperature_setpoint** | DPT_Value_Temp (9.001) | 10 | RT |
| Currently set setpoint value. | | | | |
| **DPTx_TemperatureSetpointStep** | **DP_Temperature_setpoint_MORE_LESS** | DPT_Step (1.007) | 11 | W |
| Setpoint up/down control. | | | | |
| **DPTx_HVACContrMode** | **DP_HVAC_Mode** | DPT_HVACContrMode (20.105) | 2 | RWT |
| Command for setting the operating mode. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_FanSpeedStep** | **DP_Fan_Speed_MORE_LESS** | DPT_Step (1.007) | 16 | W |
| Fan speed up/down control. | | | | |
| **DPTx_InfoFan** | **DP_Fan** | DPT_Value_1_Ucount (5.010) | 14 | RT |
| Fan speed status. | | | | |
| **DPTx_SwingVertical** | **DP_SwingVertical** | DPT_Enable (1.003) | 25 | RWT |
| Vertical swing enable status and control. | | | | |
| **DPTx_SwingHorizontal** | **DP_SwingHorizontal** | DPT_Enable (1.003) | 26 | RWT |
| Horizontal swing enable status and control. | | | | |
| **DPTx_Error** | **DP_Error** | DPT_Switch (1.001) | 27 | RT |
| Split error condition status. | | | | |
| **DPTx_ErrorCode** | **DP_Error_Code** | DPT_Value_2_Ucount (7.001) | 28 | RT |
| Error code value. | | | | |

### 4.4 Daikin via DK-RC-KNX-1/INKNXDAI001R000 with databank version 1.5



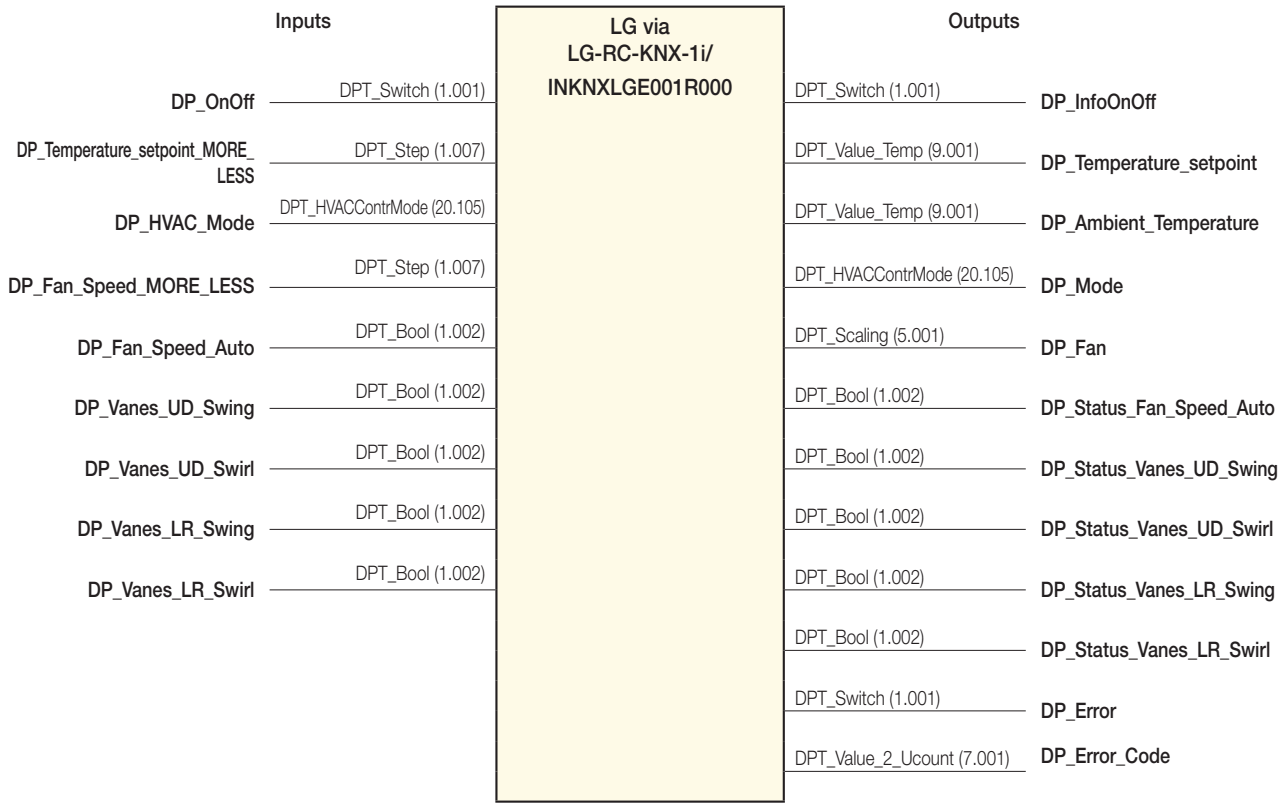

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_OnOff** | DPT_Switch (1.001) | 0 | W |
| Split on control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 47 | RT |
| Split on status. | | | | |
| **DPTx_TemperatureSetpointInfo1** | **DP_Temperature_setpoint** | DPT_Value_Temp (9.001) | 69 | RT |
| Currently set setpoint value. | | | | |
| **DPTx_TemperatureSetpointStep** | **DP_Temperature_setpoint_MORE_LESS** | DPT_Step (1.007) | 26 | W |
| Setpoint up/down control. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_AmbientTemperature** | **DP_Ambient_Temperature** | DPT_Value_Temp (9.001) | 70 | RT |
| Ambient temperature value measured by the split. | | | | |
| **DPTx_HVACContrMode** | **DP_HVAC_Mode** | DPT_HVACContrMode (20.105) | 1 | W |
| Command for setting the operating mode. | | | | |
| **DPTx_HVACContrModeInfo** | **DP_Mode** | DPT_HVACContrMode (20.105) | 48 | RT |
| Currently set operating mode status. | | | | |
| **DPTx_FanSpeedStep** | **DP_Fan_Speed_MORE_LESS** | DPT_Step (1.007) | 15 | W |
| Fan speed up/down control. | | | | |
| **DPTx_InfoFan** | **DP_Fan** | DPT_Scaling (5.001) | 56 | RT |
| Fan speed status. | | | | |
| **DPTx_VaneStep** | **DP_Vane_MORE_LESS** | DPT_Step (1.007) | 23 | W |
| Slat position up/down control. | | | | |
| **DPTx_InfoVane** | **DP_Vane** | DPT_Value_1_Ucount (5.010) | 82 | RT |
| Slat position status. | | | | |
| **DPTx_Error** | **DP_Error** | DPT_Switch (1.001) | 71 | RT |
| Split error condition status. | | | | |
| **DPTx_ErrorCode** | **DP_Error_Code** | DPT_Value_2_Ucount (7.001) | 72 | RT |
| Error code value. | | | | |

## SECTION 4: TEMPERATURE CONTROL

### 4.5 LG via LG-RC-KNX-1i/INKNXLGE001R000 with databank version 1.x




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_OnOff** | **DP_OnOff** | DPT_Switch (1.001) | 0 | W |
| Split on control. | | | | |
| **DPTx_OnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 50 | RT |
| Split on status. | | | | |
| **DPTx_TemperatureSetpointInfo1** | **DP_Temperature_setpoint** | DPT_Value_Temp (9.001) | 70 | RT |
| Currently set setpoint value. | | | | |
| **DPTx_TemperatureSetpointStep** | **DP_Temperature_setpoint_MORE_LESS** | DPT_Step (1.007) | 23 | W |
| Setpoint up/down control. | | | | |
| **DPTx_AmbientTemperature** | **DP_Ambient_Temperature** | DPT_Value_Temp (9.001) | 71 | RT |
| Ambient temperature value measured by the split. | | | | |
| **DPTx_HVACContrMode** | **DP_HVAC_Mode** | DPT_HVACContrMode (20.105) | 1 | W |
| Command for setting the operating mode. | | | | |
| **DPTx_HVACContrModeInfo** | **DP_Mode** | DPT_HVACContrMode (20.105) | 51 | RT |
| Currently set operating mode status. | | | | |
| **DPTx_FanSpeedStep** | **DP_Fan_Speed_MORE_LESS** | DPT_Step (1.007) | 17 | W |
| Fan speed up/down control. | | | | |

## SECTION 4: TEMPERATURE CONTROL

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_FanSpeedInfo** | **DP_Fan** | DPT_Scaling (5.001) | 59 | RT |
| Fan speed status. | | | | |
| **DPTx_FanSpeedAuto** | **DP_Fan_Speed_Auto** | DPT_Bool (1.002) | 12 | W |
| Automatic fan speed enable control. | | | | |
| **DPTx_FanSpeedAutoInfo** | **DP_Status_Fan_Speed_Auto** | DPT_Bool (1.002) | 60 | RT |
| Enable state. | | | | |
| **DPTx_VanesUDSwing** | **DP_Vanes_UD_Swing** | DPT_Bool (1.002) | 18 | W |
| Vanes up-down swing function enable control. | | | | |
| **DPTx_VanesUDSwingInfo** | **DP_Status_Vanes_UD_Swing** | DPT_Bool (1.002) | 66 | RT |
| Vanes up-down swing function enable status. | | | | |
| **DPTx_VanesUDSwirl** | **DP_Vanes_UD_Swirl** | DPT_Bool (1.002) | 19 | W |
| Vanes up-down swirl function enable control. | | | | |
| **DPTx_VanesUDSwirlInfo** | **DP_Status_Vanes_UD_Swirl** | DPT_Bool (1.002) | 67 | RT |
| Vanes up-down swirl function enable status. | | | | |
| **DPTx_VanesLRSwing** | **DP_Vanes_LR_Swing** | DPT_Bool (1.002) | 20 | W |
| Vanes left-right swing function enable control. | | | | |
| **DPTx_VanesLRSwingInfo** | **DP_Status_Vanes_LR_Swing** | DPT_Bool (1.002) | 68 | RT |
| Vanes left-right swing function enable status. | | | | |
| **DPTx_VanesLRSwirl** | **DP_Vanes_LR_Swirl** | DPT_Bool (1.002) | 21 | W |
| Vanes left-right swirl function enable control. | | | | |
| **DPTx_VanesLRSwirlInfo** | **DP_Status_Vanes_LR_Swirl** | DPT_Bool (1.002) | 69 | RT |
| Vanes left-right swirl function enable status. | | | | |
| **DPTx_Error** | **DP_Error** | DPT_Switch (1.001) | 72 | RT |
| Split error condition status. | | | | |
| **DPTx_ErrorCode** | **DP_Error_Code** | DPT_Value_2_Ucount (7.001) | 73 | RT |
| Error code value. | | | | |

## SECTION 5: SOUND SYSTEM

# Index

1 Introduction ..... 112
2 Overview of the sound system and basic controls ..... 112
3 RCA audio input modules ..... 113
4 IR control module ..... 113
4.1 IRModule ..... 114
4.1.1 Summary of Communication Objects ..... 114
4.1.2 Detailed description of communication Datapoints ..... 114
5 FM radio tuner with RDS ..... 116
5.1 AudioTxFMTuner ..... 116
5.1.1 Summary of Communication Objects ..... 116
5.1.2 Detailed description of communication Datapoints ..... 117
6 Bluetooth interface ..... 119
6.1 FBT_BTAUDIO ..... 119
7 Call microphone module ..... 121
7.1 AudioRockerButton ..... 121
7.1.1 “On/Off rocker” channel ..... 121
7.1.2 “Dimmer rocker” channel ..... 121
7.1.3 “Roller Shutter rocker” channel ..... 121
7.1.4 “Scenario rocker” channel ..... 121
7.1.5 “Audio On/Off Volume rocker” channel ..... 122
7.1.6 “Audio Ch+ / Track+ rocker” channel ..... 122
7.1.7 “Audio Sleep rocker” channel ..... 122
7.1.8 “Audio Monitoring rocker” channel ..... 122
7.1.9 Summary of FB1 Communication Objects ..... 122
7.1.10 Detailed description of FB1 Communication Datapoints ..... 123
7.2 AudioTxMicrophoneCall ..... 126
7.2.1 Summary of FB2 Communication Objects ..... 126
7.2.2 Detailed description of FB2 Communication Datapoints ..... 126
8 2M flush-mounting 1+1W amplifier equipped with control buttons ..... 127
8.1 AudioRockerButton ..... 127
8.2 AudioRx2MAudioOutput ..... 127
8.2.1 Summary of Communication Objects ..... 127
8.2.2 Detailed description of communication Datapoints ..... 128
9 Audio amplifier DIN rail 10+10W ..... 130
9.1 AudioRxDINAudioOutput ..... 130
10 2M 4-button home automation ampl. control 1+1 W ..... 130
10.1 FBT_BUTTON ..... 130
10.2 FBT_ROCKER BUTTON ..... 130
10.3 FBT_OUT_AUDIO ..... 130
11 4-button home automation control +1 LINEOUT 2M ..... 132
11.1 FBT_BUTTON ..... 132
11.2 FBT_ROCKER BUTTON ..... 132
11.3 FBT_OUT_AUDIO ..... 132

# SECTION 5: SOUND SYSTEM

## 1. Introduction

The document describes the features of the KNX communication interfaces of the By-me sound system devices, the method for obtaining the system configuration information and application examples for controlling and commanding system equipment via standard KNX messages.

This document does not describe the functional, technical and regulatory characteristics of the sound system, for which you should refer to the official documentation

## 2. Overview of the sound system and basic controls

The sound system is composed of input devices or transmitters, through which the desired audio content is injected into the system, and output devices or receivers, which enable listening to music content in the various environments. The By-me system commands can be configured to control the system functions.

Transmitters:

1. 2M RCA input (device Type 0x009A)
2. IR control module for 2M RCA input (device Type 0x009E)
3. FM tuner with RDS (device Type 0x009B)
4. By-me 2M Bluetooth interface (device Type 0x00C3)

Microphone Transmitters:

1. Call microphone module (device Type 0x009C)

The system has a maximum of 4 transmitters, therefore the transmission of up to 4 different simultaneous audio "channels" that can be selected by the receivers. In addition to playing audio you can make voice calls by using the call module; the system can configure a maximum of 8 microphone transmitters.

The voice call occupies one of the 4 available audio "channels" and is transmitted by the receivers involved in the call. At the end of the call the audio zones will return to transmitting the audio channel they were tuned to.

Receivers:

1. 2M flush-mounting 1+1W amplifier equipped with control buttons (device Type 0x0099)
2. 10+10W amplifier from DIN rail (device Type 0x009D)
3. 4-button home automation ampl. control 1+1W 2M (device Type 0x00C4)
4. 4-button home automation control+1 LINEOUT 2M (device Type 0x00C5)

The receiver devices are used to play the audio transmitted through one of the channels in the system. These devices are also equipped with a high quality audio amplifier allowing direct connection to the speakers.

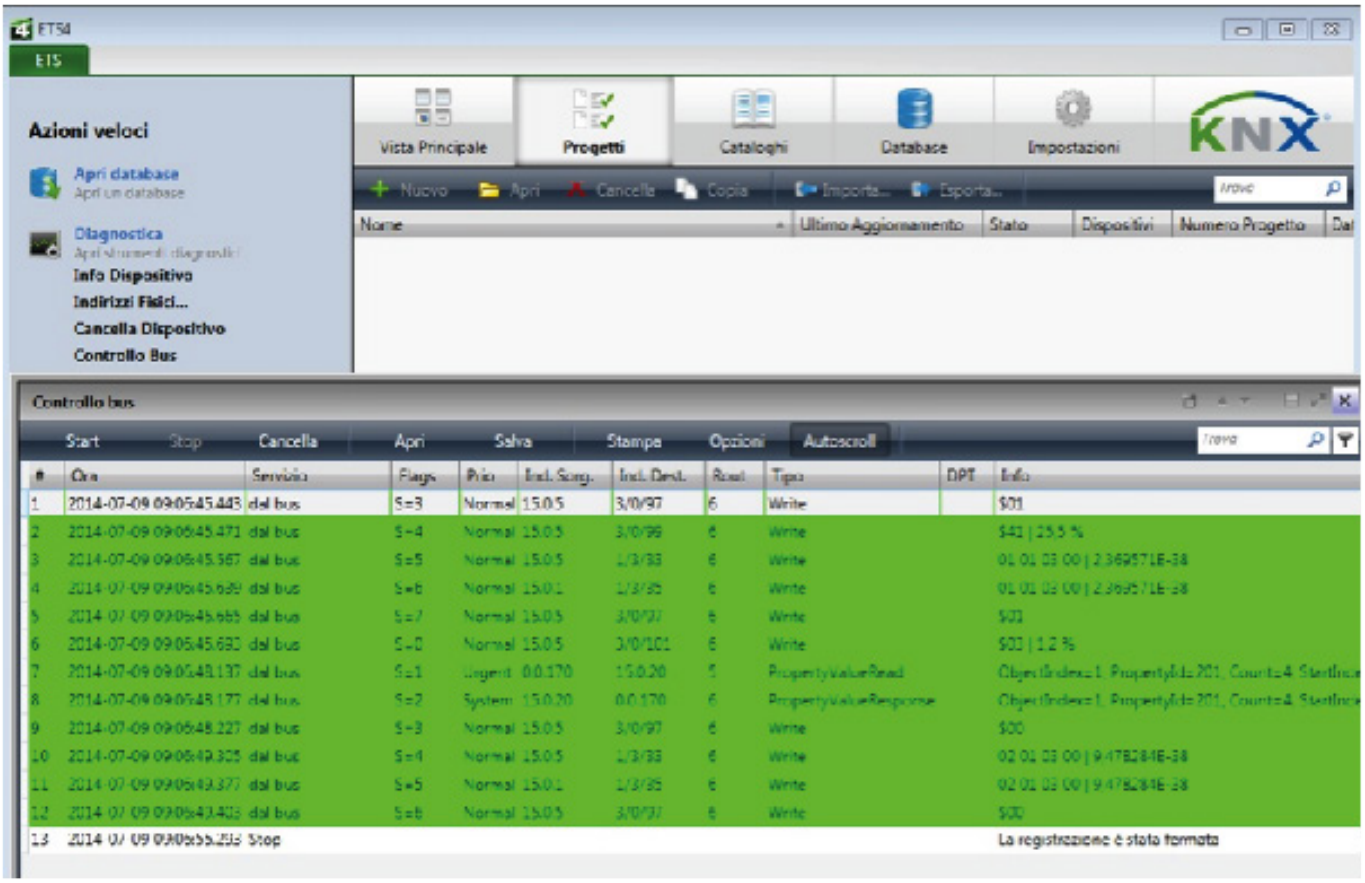
For the application configuration use the VIEW Pro app.

For each receiver, via KNX standard messages, you can command and control:

1. switch on/off (DPT 1 bit Switch)
2. volume control (DPT 4 bit Control Dimming)
3. change the received audio channel (DPT 1 bit Switch)
4. change the track or tuned FM frequency (where possible) (DPT 1 bit Switch)

Using the ETS "Bus Control" monitor, in an installed and running By-me sound system, for a KNX integrator it is easy to read the telegrams that are sent by the buttons or by more advanced systems (home automation system gateway); in this way it is possible to identify and when required replicate the commands to switch the audio zones on and off, raise and lower the volume, and change tracks and sound source.

## SECTION 5: SOUND SYSTEM




For basic applications it will then be sufficient, with the ETS "Bus Control" monitor, to obtain the group addresses and the format of the KNX standard telegrams, which then can be easily combined with other devices and/or basic supervision software functions.

For developers who want to create complete applications, the following chapters describe all the specific functions of each device, with a list and description of the available datapoints to achieve integration with other systems.

Datapoint types other than those stated herein must not be used, nor interacting on group addresses linked to datapoint types not described, as there is a risk of the system malfunctioning.

### 3. RCA audio input modules

The device consists of a single Functional Block:

| Fb Index | Fb Type |
|---|---|
| 1 | AudioTxRCAInput (0x3A) |
| The FB implements the full functionality of the device. Within the By-me system it is a transmitter for the sound system. | |

The device has no datapoint to share with the KNX system; if the RCA Input Module is combined with an IR Control Module, please refer to the communication datapoints of that device, as described in the next chapter.

### 4. IR control module

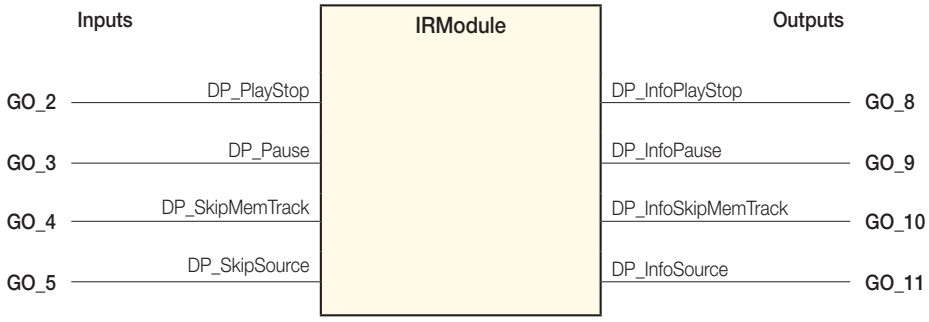
The device consists of a single Functional Block:

| Fb Index | Fb Type |
|---|---|
| 1 | IRModule (0x3E) |
| The FB implements the full functionality of the device. Within the By-me system it becomes functional only if configured in the same group as an *AudioTxRCAInput FB* made from an RCA transmitter of the sound system. It acts as a control interface of the audio source connected to the RCA transmitter with which it is paired. | |

# SECTION 5: SOUND SYSTEM

## 4.1 IRModule

The devices consist of 5 Functional Blocks:




### 4.1.1 Summary of Communication Objects

| Name in VIEW Pro | Datapoint | DPT | CO number | PID Enable | Flags |
|---|---|---|---|---|---|
| **DPTx_PlayStop** | **DP_PlayStop** | DPT_Start (1.010) | 2 | - | W |
| Commands play/stop of the device controlled by the IR. | | | | | |
| **DPTx_Pause** | **DP_Pause** | DPT_Enable (1.003) | 3 | - | W |
| Commands the pause status of the device controlled by the IR. | | | | | |
| **DPTx_IrSkipMemory** | **DP_SkipMemTrack** | DPT_Step (1.007) | 4 | - | W |
| Commands skip forward or backward in the memory or the track being played on the device controlled by the IR. | | | | | |
| **DPTx_SkipSource** | **DP_SkipSource** | DPT_Step (1.007) | 5 | - | W |
| Commands source switching, with the next or the previous one, in a multi-source device controlled by the IR. | | | | | |
| **DPTx_PlayStopInfo** | **DP_InfoPlayStop** | DPT_Start (1.010) | 8 | - | RT |
| Gives the play/stop status of the device controlled by the IR. | | | | | |
| **DPTx_PauseInfo** | **DP_InfoPause** | DPT_Enable (1.003) | 9 | - | RT |
| Gives the pause status of the device controlled by the IR. | | | | | |
| **DPTx_IrSkipMemoryInfo** | **DP_InfoSkipMemTrack** | DPT_Step (1.007) | 10 | - | RT |
| Notifies skip forward or backward in the memory or the track being played on the device controlled by the IR. | | | | | |
| **DPTx_SourceInfo** | **DP_InfoSource** | DPT_Value_1_Ucount (5.010) | 11 | - | RT |
| Gives the source selected in the multi-source device controlled by the IR. | | | | | |

### 4.1.2 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 4.1.2.1 DP_PlayStop

- Datapoint type: DPT_Start (1.010)
- Communication method: Group Object Number 2
- Flags: W

*Description*: Writing on this datapoint requires sending the Play command (paired with the value 1 = play/start) or Stop (paired with the value 0 = stop) to the device controlled by the IR module.
The IR module will send the corresponding command (cmd5 or cmd6).

#### 4.1.2.2 DP_Pause

- Datapoint type: DPT_Enable (1.003)
- Communication method: Group Object Number 3
- Flags: W

*Description*: Writing on this datapoint requires sending the Pause command (paired with the value 1 = pause) or releasing the Pause state (paired with the value 0 = resume) to the device controlled by the IR module.
The IR module will send the corresponding command (cmd7 or cmd8).

## SECTION 5: SOUND SYSTEM

#### 4.1.2.3 DP_SkipMemTrack

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 4
- Flags: W

*Description*: Writing on this datapoint requires sending the Skip Forward command (paired with the value 1 = forward) or Skip Back (paired with the value 0 = backward), in the memory or track, to the device controlled by the IR module.
The IR module will send the corresponding command (cmd3 or cmd4).

#### 4.1.2.4 DP_SkipSource

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 5
- Flags: W

*Description*: Writing on this datapoint requires sending the command for going to the next source (paired with the value 1 = forward) or the previous source (paired with the value 0 = backward) to the device controlled by the IR module. By source we mean an internal module of a multi-source device, eg. a compact unit with FM tuner, CD and MC player.
The IR module will send the corresponding command to the selected source (cmd9, cmd10 or cmd11).

#### 4.1.2.5 DP_InfoPlayStop

- Datapoint type: DPT_Start (1.010)
- Communication method: Group Object Number 8
- Flags: RT

*Description*: The message is sent following a request to play or stop. It is basically feedback (Response) on the writing on DP_PlayStop. Therefore the person sending the command can check if the play or stop message was successful. The message can also be requested on reading at any time and therefore not in relation to writing on DP_PlayStop. This reading does not give the status of the Audio source, but the last command sent to the source (paired with the value 1 = play/start or paired with the value 0 = stop).

#### 4.1.2.6 DP_InfoPause

- Datapoint type: DPT_Enable (1.003)
- Communication method: Group Object Number 9
- Flags: RT

*Description*: The message is sent following a request for Info or Pause. It is basically feedback (Response) on the writing on DP_Pause. Therefore the person sending the command can check if the info or pause message was successful. The message can also be requested on reading at any time and therefore not in relation to writing on DP_Pause. This reading does not give the status of the Audio source, but the last command sent to the source (paired with the value 1 = pause or with the value 0 = resume).

#### 4.1.2.7 DP_InfoSkipMemTrack

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 10
- Flags: RT

*Description*: The message is sent following a request to Skip the memory or track. It is basically feedback (Response) on the writing on DP_SkipMemTrack. Therefore the person sending the command can check if the skip message was successful. The message can also be requested on reading at any time and therefore not in relation to writing on DP_SkipMemTrack. This reading does not give the status of the Audio source, but the last command sent to the source (1 = forward or 0 = backward).

#### 4.1.2.8 DP_InfoSource

- Datapoint type: DPT_Value_1_Ucount (5.010)
- Communication method: Group Object Number 11
- Flags: RT

*Format*: this identifies the index of the source it is paired to, according to the following list

| | |
|---|---|
| 0x00. | No source |
| 0x01. | Source 1 |
| 0x02. | Source 2 |
| 0x03. | Source 3 |
| 0x04-0xFF | not used |

*Description*: The message is sent following a request to change source. It is basically feedback (Response) on the writing on DP_SkipSource. Therefore the person sending the command can check if the skip source message was successful. The message can also be requested on reading at any time and therefore not in relation to writing on DP_SkipSource. This reading gives the value of the Audio source selected by the last command.

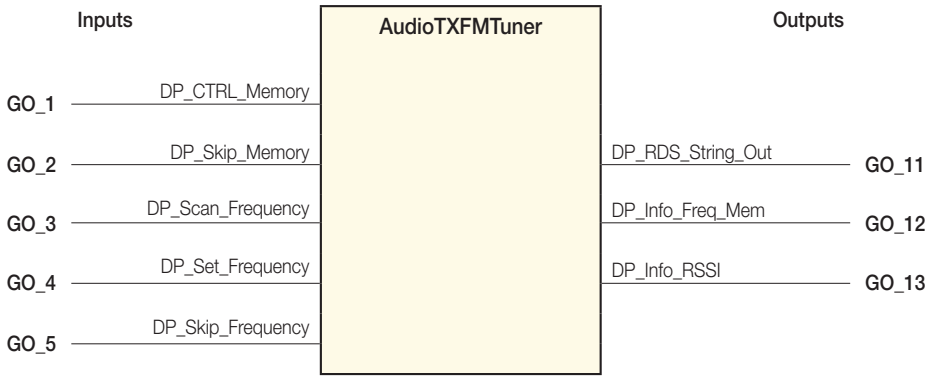
# SECTION 5: SOUND SYSTEM

## 5. FM radio tuner with RDS

The device consists of a single Functional Block:

| Fb Index | Fb Type |
|---|---|
| 1 | AudioTxFMTuner (0x3B) |
| The FB implements the full functionality of the device. Within the By-me system it is a transmitter for the sound system. Using the implemented Communication Objects it can fully control the paired FM tuner. | |

### 5.1 AudioTxFMTuner




#### 5.1.1 Summary of Communication Objects

| Name in VIEW Pro | Datapoint | DPT | CO number | PID Enable | Flags |
|---|---|---|---|---|---|
| **DPTx_MemoryControl** | **DP_CTRL_Memory** | Vimar Proprietors | 1 | - | W |
| Saving and calling up a radio station by memory index | | | | | |
| **DPTx_SkipMemory** | **DP_ Skip_Memory** | DPT_Step (1.007) | 2 | - | W |
| Switches to the radio station saved with the previous or next index | | | | | |
| **DPTx_FrequencyScan** | **DP_ Scan_Frequency** | DPT_Step (1.007) | 3 | - | W |
| Starts automatically scanning the next or previous frequencies, in relation to the first one | | | | | |
| **DPTx_FrequencySet** | **DP_ Set_Frequency** | Vimar Proprietors | 4 | - | W |
| Sets a specific frequency value | | | | | |
| **DPTx_FrequencySkip** | **DP_ Skip_Frequency** | DPT_Control_Dimming (3.007) | 5 | - | W |
| Increases or decreases the tuning frequency with a minimum step of 50 kHz. The control is the dimming type, therefore continuous changes can be started, until the stop command is sent. | | | | | |
| **DPTx_RdsName** | **DP_RDS_String_Out** | Vimar Proprietors | 11 | - | RT |
| Gives the string for the RDS information sent by the radio station currently tuned into. | | | | | |
| **DPTx_FrequencyMemoryInfo** | **DP_Info_Freq_Mem** | Vimar Proprietors | 12 | - | RT |
| Gives the value of the frequency and the memory index, if any, of the currently tuned radio station | | | | | |
| **DPTx_RssiInfo** | **DP_Info_RSSI** | DPT_Percent_U8 (5.004) | 13 | - | RT |
| Gives an approximate value of the quality of the radio signal received by the FM tuner | | | | | |

# SECTION 5: SOUND SYSTEM

### 5.1.2 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below

#### 5.1.2.1 DP_CTRL_Memory

- Datapoint type: Vimar Proprietors
- Communication method: Group Object Number 1
- Flags: W

*Format*: 1 byte

| Bit 7 | Bit 6 - Bit 4 | Bit 3 - Bit 0 |
|---|---|---|
| Saving | Not Used | Memory index |

Saving {0, 1} = saved command code or memory call-up
1. Saves the currently tuned station
0. Calls up a previously saved station

Memory index [1..8] = index of the memory to be called up or paired with the station to save

*Description*: The message comes from any device that can send a 1 byte message. When calling up a memory, the transmitter will need to tune the frequency paired with the communicated memory index. Whereas, when saving, the transmitter will need to save the current frequency and pair it with the communicated memory index, so as to then be able to call it up starting from the same index.

#### 5.1.2.2 DP_Skip_Memory

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 2
- Flags: W

*Description*: Writing on this datapoint enables tuning the radio station paired with the previous memory index (data field with value 0 = previous) or the next one (data field with value 1 = next) in relation to the currently selected index.

#### 5.1.2.3 DP_Scan_Frequency

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 3
- Flags: W

*Description*: Writing on this datapoint enables starting automatic scanning for radio stations at the lower frequencies (data field with value 0 = scan down) or higher ones (data field value 1 = scan up) in relation to the currently tuned frequency. The transmitter must then search for the next or previous station with a greater signal level (SNR ratio) than the percentage specified by the parameter "PAR_Search_Threshold". If a new station is found, the transmitter will tune onto this one, otherwise it will continue scanning the entire band until it returns to the original frequency. The time required to scan the entire band (and so if no frequency is found with the SNR above the required level) is approximately 23 seconds.

#### 5.1.2.4 DP_Set_Frequency

- Datapoint type: Vimar Proprietors
- Communication method: Group Object Number 4
- Flags: W

*Format*: 2 bytes

| Byte 0 | Byte 1 |
|---|---|
| MSB_Frequency | LSB_Frequency |

Frequency [8750.10800] = Frequency to be tuned expressed in tens of kHz, i.e.:
8750 = 87.50 MHz
8755 = 87.55 MHz
…
10800 = 108.00 MHz

*Description*: Writing on this datapoint enables the setting of a specific frequency to be tuned to on the FM radio. The value written on the 2 bytes of the datapoint is none other than the frequency expressed in MHz (with no more than 2 decimal places) and multiplied by 100.

## SECTION 5: SOUND SYSTEM

#### 5.1.2.5 DP_Skip_Frequency

- Datapoint type: Vimar Proprietors
- Communication method: Group Object Number 5
- Flags: W

*Format*: 4 bytes

| Bit 7 - Bit 4 | Bit 3 | Bit 2 - Bit 0 |
|---|---|---|
| Not Used | C | StepCode |

C {0, 1} = Control, that is it gives the direction of the change in frequency:
0. Decreases the frequency to tune to
1. Increases the frequency to tune to

StepCode [0..7] = Minimum step with which to vary the frequency
0. STOP change
1. START change (fixed step of 50 kHz)
[2..7] RESERVED

*Description*: On receiving a message with StepCode equal to 1 the FM radio tuner starts increasing or decreasing (depending on the value of the Control bit, c = 1/0) the value of the next frequency to tune to. Only when the device receives a message with StepCode equal to 0 will the changing frequency stop and the corresponding radio station be tuned. Initially the frequency changes slowly, then progressively more quickly, so as to be able to generate an acceleration in the increase/decrease when the command remains active longer. In this way, by pressing briefly you can change a single frequency, while pressing for longer lets you quickly make broader changes in frequency. The minimum step for changing frequency is 50 kHz.

#### 5.1.2.6 DP_RDS_String_Out

- Datapoint type: Vimar Proprietors
- Communication method: Group Object Number 11
- Flags: RT

*Format*: 8 bytes containing the 8 characters of each RDS string not terminated by byte '\0'.

*Description*: The message is sent to let the graphic interfaces display the information received from the tuned radio station. The data to be sent on the KNX bus comes directly from the FM tuner, but since there is no type of protection (such as CRC or checksum) on the data sent by radio stations, it is necessary to somehow check the accuracy of the data received, before sending it on the bus.
The format of the sent data consists of 8 bytes corresponding to 8 characters that make up the PS (Programme Service) data. The codes to use to decode each character sent are given in Table E.1 (Basic RDS character set), specified in the standard IEC 62106:2009 – Annex E (page 72).

#### 5.1.2.7 DP_Info_Freq_Mem

- Datapoint type: Vimar Proprietors
- Communication method: Group Object Number 12
- Flags: RT

*Format*: 3 bytes

| Byte 0 | Byte 1 | Byte 2 |
|---|---|---|
| MSB_Frequency | LSB_Frequency | Memory index |

Frequency [8750.10800] = Frequency to be tuned expressed in tens of kHz, i.e.:
8750 = 87.50 MHz
8755 = 87.55 MHz
…
10800 = 108.00 MHz

Memory index [0..8] = index of the memory paired with the tuned frequency
0 = tuned frequency not in memory
1..8 = memory index paired with the tuned frequency and previously saved
9..255 = RESERVED

*Description*: The message is sent on a change in the set frequency. It therefore contains information on the frequency just tuned and the corresponding memory index, if it is a station already in memory, otherwise the memory index field will be 0. This message is also sent after saving a new frequency.

## SECTION 5: SOUND SYSTEM

#### 5.1.2.8 DP_ Info_ RSSI

- Datapoint type: DPT_Percent_U8 (5.004)
- Communication method: Group Object Number 13
- Flags: RT

*Format*: 1 byte

| Bit 7 | Bit 6 - Bit 0 |
|---|---|
| Not Used | RSSI_% |

RSSI_% [0..100] = Percentage of RSSI
0x00 = 0 %
…
0x64 = 100 %

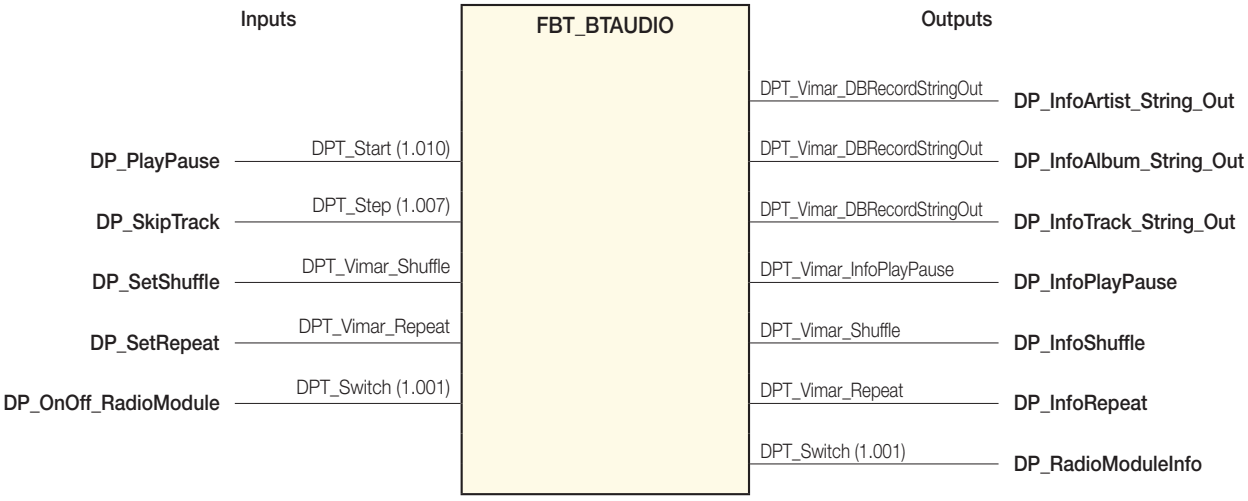
*Description*: The message is sent to provide information on the quality of the input signal received by the FM tuner: from 0% (very disturbed signal) to 100% (excellent signal). It is sent after each change in the detected RSSI value, with each tuning of a new frequency or also when expressly requested by some other device.

## 6. Bluetooth interface

The device consists of a single Functional Block:

| Fb Index | Fb Type |
|---|---|
| 1 | FBT_BTAUDIO (0x6B) |

### 6.1 FBT_BTAUDIO




| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_PlayPause** | **DP_PlayPause** | DPT_Start (1.010) | 3 | W |
| Controls the device play(1) or pause (0) received via BT. | | | | |
| **DPTx_SkipTrack** | **DP_SkipTrack** | DPT_Step (1.007) | 4 | W |
| Writing on this datapoint requires the sending of the command to play the previous track (paired to the value 0 = previous) or the next one (paired to the value 1 = next ) received via BT. | | | | |

## SECTION 5: SOUND SYSTEM

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_SetShuffle** | **DP_SetShuffle** | DPT_Vimar_Shuffle | 5 | W |
| Writing on this datapoint requires the sending of a command to enable or disable the shuffle track playing mode on the device received via BT<br><br>Shuffle_Mode [0..3] = command code<br>00. RESERVED<br>01. SHUFFLE_OFF<br>10. SHUFFLE_TRACKS<br>11. NO_ACTION<br><br>Not being able to guarantee the function for all smartphones on the market, the device does not manage the datapoint which however remains in the model. | | | | |
| **DPTx_SetRepeat** | **DP_SetRepeat** | DPT_Vimar_Repeat | 6 | W |
| Writing on this datapoint requires the sending of a command to enable or disable the repeat playing of the tracks on the device received via BT<br><br>Repeat_Mode [0..3] = command code<br>00. RESERVED<br>01. REPEAT_OFF<br>10. REPEAT_ONE_TRACK<br>11. REPEAT_ALL_TRACKS | | | | |
| **DPTx_OnOffRadioModule** | **DP_OnOff_RadioModule** | DPT_Switch (1.001) | 7 | W |
| Message received as a request to switch on the BT radio module, not to switch on the device. | | | | |
| **DPTx_ArtistName** | **DP_InfoArtist_String_Out** | DPT_Vimar_DBRecordStringOut | 9 | RT |
| The message is sent to offer the possibility for the graphic interfaces to view the information concerning the name of the artist paired to the track being played. | | | | |
| **DPTx_AlbumName** | **DP_InfoAlbum_String_Out** | DPT_Vimar_DBRecordStringOut | 10 | RT |
| The message is sent to offer the possibility for the graphic interfaces to view the information concerning the name of the album paired to the track being played. | | | | |
| **DPTx_TrackName** | **DP_InfoTrack_String_Out** | DPT_Vimar_DBRecordStringOut | 11 | RT |
| The message is sent to offer the possibility for the graphic interfaces to view the information on the name of the track being played. | | | | |
| **DPTx_InfoPlayPause** | **DP_InfoPlayPause** | DPT_Vimar_InfoPlayPause | 12 | RT |
| Gives the play status of the device received via BT. | | | | |
| **DPTx_InfoShuffle** | **DP_InfoShuffle** | DPT_Vimar_Shuffle | 13 | RT |
| Gives the shuffle status of the device received via BT. | | | | |
| **DPTx_InfoRepeat** | **DP_InfoRepeat** | DPT_Vimar_Repeat | 14 | RT |
| The message is sent after each change of the track repeat mode: in particular, the playing ends with the last track in the list (Repeat Off) or starts again once the playlist being played has ended (Repeat all tracks) or continuously repeats the same track (Repeat one track). | | | | |
| **DPTx_InfoOnOffRadioModule** | **DP_RadioModuleInfo** | DPT_Switch (1.001) | 15 | RT |
| The message is sent after a change in the BT radio module status, not the device. | | | | |

# SECTION 5: SOUND SYSTEM

## 7. Call microphone module

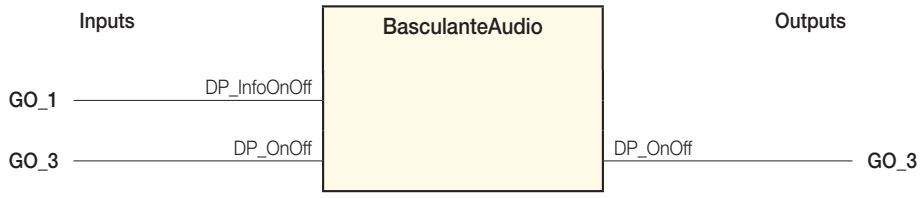
The device consists of two Functional Blocks:

| Fb Index | Fb Type |
|---|---|
| 1 | BasculanteAudio |
| The FB implements a generic button referring to the RH rocker button on the device. Configurable as Rocker, Dimmer, Roller Shutter, Scenario Recall, Audio (On/Off Volume, Ch+/Track+, Sleep) or Audio Monitoring. | |
| 2 | AudioTxMicrophoneCall (0x3C) |
| The FB implements the primary function of the device, that is performing the General or Selective call services to transmit the sounds captured by the front microphone. The ADJ Select parameter paired with this FB identifies channels linked to different application logic elements geared to the dynamic control of the call channel. | |

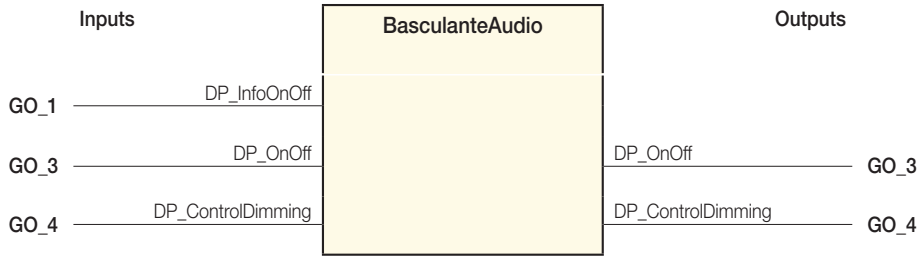
### 7.1 Audio Rocker

Depending on the group in which it is learnt the Audio Rocker button can take the following application channel types.

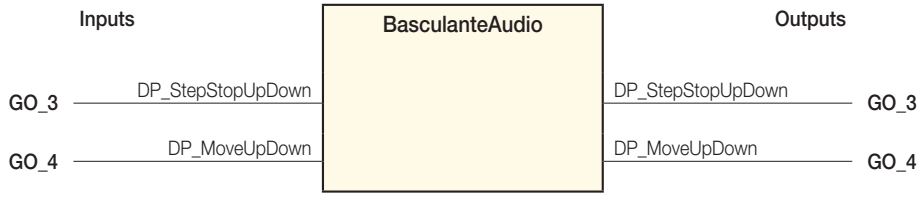
#### 7.1.1 “On/Off rocker” channel



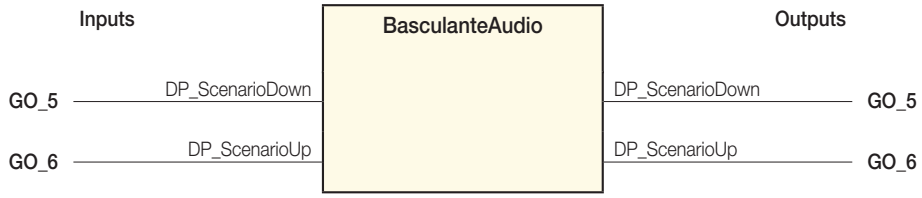

#### 7.1.2 “Dimmer rocker” channel




#### 7.1.3 “Roller Shutter rocker” channel

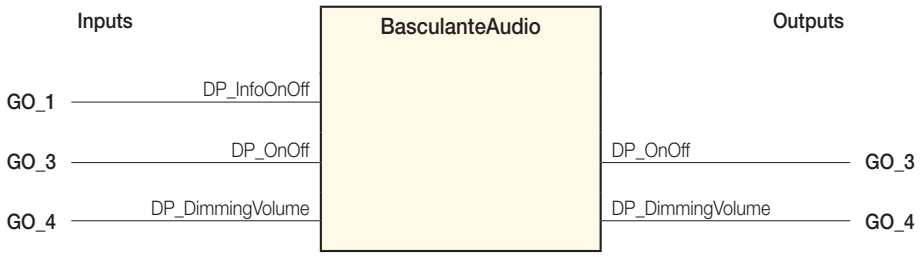



#### 7.1.4 “Scenario rocker” channel

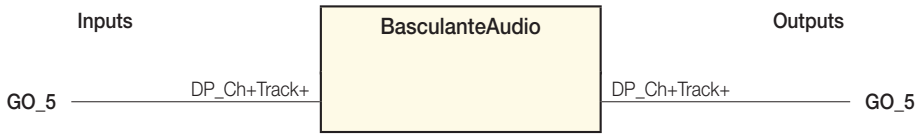


## SECTION 5: SOUND SYSTEM

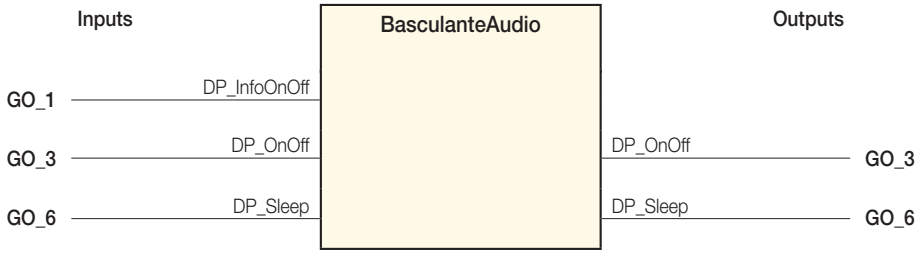
### 7.1.5 “Audio On/Off Volume rocker” channel



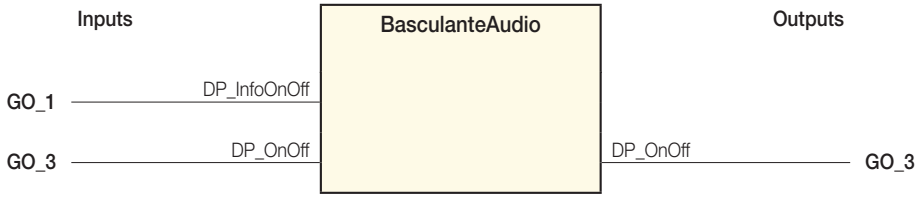

### 7.1.6 “Audio Ch+ / Track+ rocker” channel




### 7.1.7 “Audio Sleep rocker” channel




### 7.1.8 “Audio Monitoring rocker” channel




### 7.1.9 Summary of FB1 Communication Objects

| Name in VIEW Pro | Datapoint | DPT | CO number | PID Enable | Flags |
|---|---|---|---|---|---|
| **DPTx_OnOffInfo *** | **DP_Info OnOff** | DPT_Switch (1.001) | 1 | - | W |
| The message is received at the input following a change in the status of the paired control (relay, dimmer, rx audio), i.e. every time it passes from On to Off and vice versa. | | | | | |
| **DPTx_OnOff *** | **DP_OnOff** | DPT_Switch(1.001) | 3 | - | WT |
| Controls activating/deactivating the paired actuator (relay, dimmer, receiver.. depending on the channel) | | | | | |
| **DPTx_StopStepUpDown** | **DP_StepStopUpDown** | DPT_Switch(1.001) | 3 | - | WT |
| The message is sent to control stopping the previous UP or DOWN actuation (via DP_MoveUpDown) of the paired roller shutter actuator | | | | | |
| **DPTx_UpDown** | **DP_MoveUpDown** | DPT_UpDown (1.008) | 4 | - | WT |
| Controls the paired roller shutter actuator Up or Down | | | | | |
| **DPTx_DimmingControl** | **DP_ControlDimming** | DPT_ControlDimming (3.007) | 4 | - | T |
| Message used to control a dimmer actuator in the automation system. | | | | | |

## SECTION 5: SOUND SYSTEM

| Name in VIEW Pro | Datapoint | DPT | CO number | PID Enable | Flags |
|---|---|---|---|---|---|
| **DPTx_VolumeControl** | **DP_DimmingVolume** | DPT_ControlDimming (3.007) | 4 | - | T |
| Message sent to control increasing or decreasing the volume of the receivers of the paired audio group. | | | | | |
| **DPTx_SceneActivator1** | **DP_ScenarioDown** | DPT_SceneControl (18.001) | 5 | - | T |
| Message sent to control the scenario corresponding to the PAR_ScenarioDown parameter | | | | | |
| **DPTx_SkipChannelTrack** | **DP_Ch+/Track+** | Vimar Proprietors | 5 | - | T |
| EIS1 Message corresponding to the next channel (top button Ch+) and next track (bottom button Track+) functions of the paired audio receivers. | | | | | |
| **DPTx_SceneActivator2** | **DP_ScenarioUp** | DPT_SceneControl (18.001) | 6 | - | T |
| Message sent to control the scenario corresponding to the PAR_ScenarioUp parameter | | | | | |
| **DPTx_Sleep** | **DP_Sleep** | DPT_Enable (1.003) | 6 | - | WT |
| Message to activate the Sleep function of the audio receivers of the zones paired with the button. | | | | | |

#### 7.1.10 Summary of FB1 Communication Objects

The detailed descriptions of each Datapoint covered by the model are given below.

##### 7.1.10.1 DP_Info OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: W

*Format*: 1 bit (1= ON, 0= OFF)

*Description*: the message is received at the input following a change in the status of the paired control (relay, dimmer, rx audio), i.e. every time it passes from On to Off and vice versa.

##### 7.1.10.2 DP_OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 3
- Flags: WT

*Format*: 1 bit (1= ON, 0= OFF)

*Description*: the message is associated with briefly pressing the top and bottom buttons of the rocker and sent as an on/off command of the corresponding actuator (relay, dimmer in Automation mode), audio zone or audio monitoring service implemented by the call module transmitter. It is also received on input following an On/Off of another actuator in the same group (for functions relating to the control of the front LEDs on the rocker button).

##### 7.1.10.3 DP_StepStopUpDown

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 3
- Flags: T

*Format*: 1 bit (1= Stop UP → briefly pressing the top button, 0= Stop DOWN → briefly pressing the bottom button).
The message is then of type **E1 00 8X** with X which, depending on the requests, takes the following values:

| (X value) | Action |
|---|---|
| 1 | Roller Shutter Up Stop (briefly pressing top button) |
| 0 | Roller Shutter Down Stop (briefly pressing top button) |

*Description*: the message is sent (output DP) to control stopping the previous UP or DOWN actuation (DP Move UpDown) for a roller shutter actuator. The message therefore is meaningful only when the button is configured as "Roller Shutter Control".

## SECTION 5: SOUND SYSTEM

#### 7.1.10.4 DP_MoveUpDown

- Datapoint type: DPT_UpDown (1.008)
- Communication method: Group Object Number 4
- Flags: T

*Format*: 1 bit (1= DOWN, 0= UP)
The message is then of type **E1 00 8X** with X which, depending on the requests, takes the following values:

| (X value) | Action |
|---|---|
| 0 | Roller Shutter Up (pressing and holding top button) |
| 1 | Roller Shutter Down (pressing and holding bottom button) |

*Description*: the message is sent (DP output) to command the roller shutter actuator UP (press and hold the top button) or DOWN (press and hold the bottom button). The latter notifies the action performed by issuing a Move UpDown Info message received by the button on the previous GO1) Info MoveUpDown. The message therefore is meaningful only when the button is configured as "Roller Shutter Control".

#### 7.1.10.5 DP_ControlDimming

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 4
- Flags: T

*Format*: 4 bit - b3: 1 = UP (increase, top button) and 0 = Down (decrease, bottom button)
-b2,b1,b0 = stepCode (001b fixed in By-me to start dimming, 000b to stop dimming)
The message is then of type **E1 00 8X** with X which, depending on the requests, takes the following values:

| b3..b0 (X value) | Action |
|---|---|
| 9 | Dimming Up (pressing and holding top button) |
| 8 | Stop Dimming Up (releasing top button) |
| 1 | Dimming Down (pressing and holding bottom button) |
| 0 | Stop Dimming Down (releasing bottom button) |

*Description*: the message is sent (DP output) to control a dimmer actuator in the automation system. The dimmer actuator will run the dimming command only if it is already ON (previous short press on top On/Off command button on GO3). The message therefore is meaningful only when the button is configured as "Dimmer Control" in Automation mode.

#### 7.1.10.6 DP_DimmingVolume

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 4
- Flags: T

*Format*: 4 bit:
- b3: 1 = UP (increase, top button) and 0 = Down (decrease, bottom button)
-b2,b1,b0 = stepCode (001b fixed in By-me to start dimming, 000b to stop dimming)

Note: Similarly to the "Control Dimming" DP, the message is of type E1 00 8X with X which, depending on the requests, takes the following values:

| b3..b0 (X value) | Action |
|---|---|
| 9 | Start volume increase (pressing and holding top button) |
| 8 | Stop volume increase (releasing top button) |
| 1 | Start volume decrease (pressing and holding bottom button) |
| 0 | Stop volume decrease (releasing bottom button) |

*Description*: the message is sent (DP out) to control the action of increasing or decreasing the volume of the audio receivers when the button is configured to operate in the "Vimar Audio" application in OnOff/Volume mode.

## SECTION 5: SOUND SYSTEM

#### 7.1.10.7 DP_ScenarioDown

- Datapoint type: DPT_SceneControl (18.001)
- Communication method: Group Object Number 5
- Flags: T

*Format*: 8 bit:

- b7: 0 = scenario on
- b6..b0 = scenario index corresponding to the Down scenario parameter (values from 1 to 63).

*Description*: the message is sent (DP out) to control the action of increasing or decreasing the volume of the audio receivers when the button is configured to operate in the "Vimar Audio" application in OnOff/Volume mode.

#### 7.1.10.8 DP_Ch+_Track+

- Datapoint type: Vimar Proprietors
- Communication method: Group Object Number 5
- Flags: T

*Format*: 1 bit:

- b0=1 → pressing Top button (Ch+: change Channel)
- b0=0 → pressing Bottom button (Track+: change track)

*Description*: the message is generated (out DP) on pressing the corresponding button when it is configured to operate in the "Vimar Audio" application in "change channel/track" mode.
On pressing the top button an On command is sent (equivalent to Ch+ for increasing channel), on pressing the bottom button an Off command is sent (equivalent to Track+ for increasing track).
When the Ch+ message is received the receiver must perform the registration phase on the channel following the current one (after checking the settings of the "Active Channels" configuration parameter) through the "Tx GenericCommand" datapoint. If the procedure is successful then the system tunes to that channel, otherwise it goes on to the next channel.
On the other hand, if the command reaching the receiver is Track+, this must forward the skip track request to the Tx of the channel to which it is connected via the "Tx Generic Command" datapoint.

#### 7.1.10.9 DP_ScenarioUp

- Datapoint type: DPT_SceneControl (18.001)
- Communication method: Group Object Number 6
- Flags: T

*Format*: 8 bit:

- b7: 0 = scenario on
- b6..b0 = scenario index corresponding to the Down scenario parameter (values from 1 to 63).

*Description*: the message is sent to activate a scenario corresponding to the ScenarioUp parameter of the functional block. The message therefore is meaningful only when the button is configured in the automation domain as "scenario activator".

#### 7.1.10.10 DP_Sleep

- Datapoint type: DPT_Enable (1.003)
- Communication method: Group Object Number 6
- Flags: WT

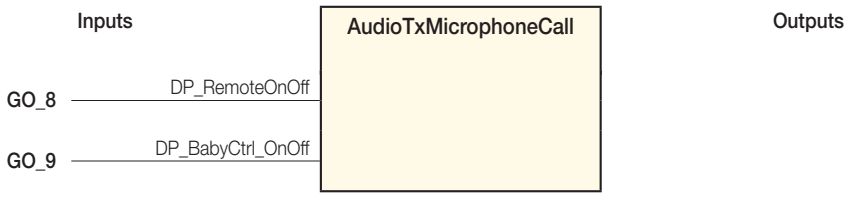
*Format*: 1 bit:

- b0 = 1 → ON (Sleep ON, Top button)
- b0 = 0 → OFF (Sleep OFF, Bottom button)

*Description*: The message is sent (DP out), on pressing the buttons, to the receivers of the audio zone for which the button is configured which must switch on on the last channel in which they were tuned for a time determined by the "Sleep Time" configuration parameter. Upon receiving the OFF message the receiver must switch off instantly.
Each time the channel or the receiver volume is changed, during Sleep time, or a new Sleep message arrives with ON, the Sleep time must be reloaded.

## SECTION 5: SOUND SYSTEM

### 7.2 AudioTxMicrophoneCall




#### 7.2.1 Summary of FB2 Communication Objects

| Names in VIEW Pro | Datapoint | DPT | CO number | PID Enable | Flags |
|---|---|---|---|---|---|
| **DPTx_GeneralCallInfo** | **DP_Remote_OnOff** | DPT_Switch (1.001) | 8 | - | WR |
| The message controls activation/deactivation of the Audio Monitoring service of the device | | | | | |
| **DPTx_SelectiveCallInfo** | **DP_BabyCtrl_OnOff** | DPT_Switch (1.001) | 9 | - | WR |
| The message controls activation/deactivation of the BabyControl service of the device | | | | | |

#### 7.2.2 Detailed description of FB2 Communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

##### 7.2.2.1 DP_Remote_OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 8
- Flags: WR

*Format*: 1 bit (1 = Audio Monitoring ON, 0 = Audio Monitoring OFF)

*Description*: This message activates/deactivates the "Audio Monitoring" service of the device. The message comes from any device capable of sending an On/Off command (typically a configured button in the same audio zone, control panel or touchscreen). On receiving the On command, the transmitter, after suitably setting the core audio, must perform a selective call request (on datapoint 12 out) indicating the audio zones to receive the "Audio Monitoring" service via the relative "Mask Audio Monitoring Zone" parameter. Then it can begin transmitting on the channel corresponding to the "Channel" parameter (or negotiated dynamically) and end on Off.

##### 7.2.2.2 DP_BabyCtrl_OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 9
- Flags: WR

*Format*: 1 bit

- 1 = ON (enables Baby Control function)
- 0 = OFF (disabled Baby Control function)

*Description*: The message commands the remote "Baby Control" service on/off similarly to local control via the front button. This message comes from any advanced interface (control panel or touchscreen) able to send a "Baby Control On/Off" command. On receiving the On command, the transmitter, after appropriately setting the parameters of the core audio (squelch threshold in particular), begins monitoring the signal from the microphone source. On exceeding the squelch threshold, the transmitter is activated, emitting a selective call request (on datapoint 12 out) indicating the audio zones to receive the "Baby Control" service via the relative "Mask Baby Control Zone" parameter. The function is switched off with EIS1 OFF on this DataPoint or on disabling via the front button.

# SECTION 5: SOUND SYSTEM

## 8. 2M flush mounting 1+1W amplifier equipped with control buttons

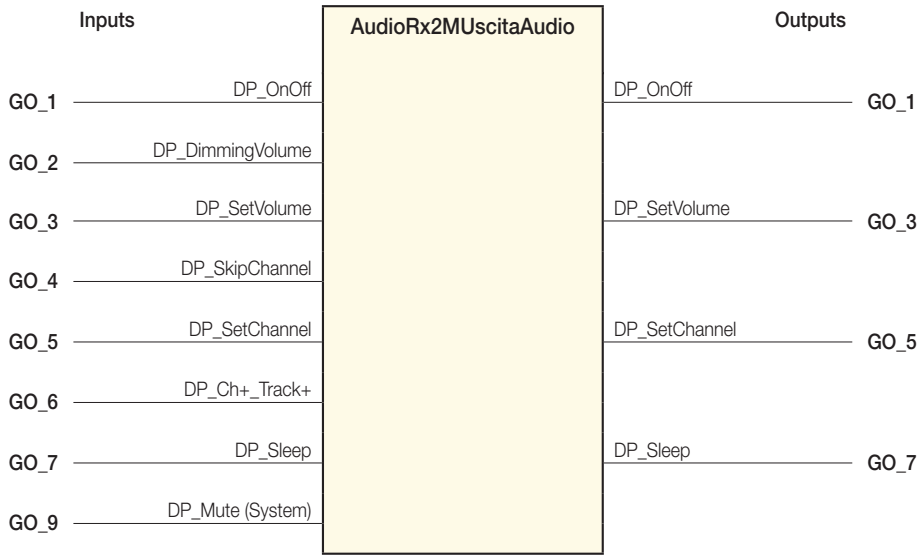
The device contains the following FB, allocated in the position indicated in the table:

| Fb Index | Fb Type |
|---|---|
| 1 | BasculanteAudio |
| The FB implements a generic button referring to the LH rocker button on the device. Configurable as Rocker, Dimmer, Roller Shutter, Scenario Recall, On/Off Volume, Ch+/Track+, Sleep or Audio Monitoring. | |
| 2 | BasculanteAudio |
| The FB implements a generic button referring to the RH rocker button on the device. Configurable as Rocker, Dimmer, Roller Shutter, Scenario Recall, On/Off Volume, Ch+/Track+, Sleep or Audio Monitoring. | |
| 3 | AudioRx2MUscitaAudio |
| Audio output for controlling 2 1W speakers. | |

### 8.1 Audio Rocker

For a description of the communication datapoints please refer to the "Audio Rocker" section in the previous chapter "call microphone module", since the functional blocks have the same model.

### 8.2 AudioRx2MUscitaAudio




#### 8.2.1 Summary of Communication Objects

| Names in VIEW Pro | Datapoint | DPT | CO number | PID Enable | Flags |
|---|---|---|---|---|---|
| **DPTx_AudioOnOff** | **DP_OnOff** | DPT_Switch (1.001) | 1 | - | RWT |
| Message received as a request to switch on the zone, while it is used at the output to repeat the message to the slaves. | | | | | |
| **DPTx_VolumeControl** | **DP_DimmingVolume** | DPT_ControlDimming (3.007) | 2 | - | W |
| Message received as a request to increase or decrease the receiver volume. | | | | | |
| **DPTx_VolumeValue** | **DP_SetVolume** | DPT_Percent_U8 (5.004) | 3 | - | RWT |
| Message received as a request to set the volume, while it is used to repeat the volume value to the slave receivers. | | | | | |
| **DPTx_AudioChSkipWithdrawn** | **DP_SkipChannel** | DPT_Step(1.007) | 4 | - | W |
| Message received as a request to change the listening channel. | | | | | |

## SECTION 5: SOUND SYSTEM

| Names in VIEW Pro | Datapoint | DPT | CO number | PID Enable | Flags |
|---|---|---|---|---|---|
| **DPTx_Channel** | **DP_SetChannel** | DPT_Value_1_Ucount(5,010) | 5 | - | RWT |
| Message received as a request to set the channel, while it is used to repeat the channel value to the slave receivers. | | | | | |
| **DPTx_SkipChannelTrack** | **DP_Ch+/Track+** | Vimar Proprietors | 6 | - | W |
| Message received as a request to increase the channel and increase the track. | | | | | |
| **DPTx_Sleep** | **DP_Sleep** | DPT_Enable (1.003) | 7 | - | RWT |
| Message received as a sleep request, i.e. for switching on for a time equal to the Sleep Time parameter, while it is used to repeat the sleep request to the slave receivers. | | | | | |
| **DPTx_SysAudioMute** | **DP_Mute** | DPT_Enable (1.003) | 9 | - | W |
| Message received for setting the receiver volume to the value of the PAR_VolumeMute parameter | | | | | |

### 8.2.2 Detailed description of the communication Datapoints

The detailed descriptions of each Datapoint covered by the model are given below.

#### 8.2.2.1 DP_OnOff

- Datapoint type: DPT_Switch (1.001)
- Communication method: Group Object Number 1
- Flags: RWT

*Description*: The message comes from any device that can send an On/Off command. On receiving an On command, after registration, the receiver must switch on on the last channel it was tuned to with the previous volume if the volume is within an acceptable window defined during configuration. When switching off it must save the current channel and the current volume.
On receiving a groupValueWrite with a value outside the range [0..1] the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected).

**N.B.: It is used as an output to repeat the message to the Slave receivers as an alternative to the On/Off Info datapoint. It is used as an output if the "PAR_AdjustableSelect" parameter is set to Vimar(=0) and the receiver is the Master Group.**

#### 8.2.2.2 DP_DimmingVolume

- Datapoint type: DPT_Control_Dimming (3.007)
- Communication method: Group Object Number 2
- Flags: W

*Description*: The message comes from any device that can send the dimmer command message.
The message is of type E1 00 8X with X varying in the different cases of requests to increase or decrease the volume.

The format of the command is as follows:

| 1 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Bit 0-3 = 9 → Start up, 8 → Stop up, 1 → Start Down, 0 → Stop Down

On receiving a groupValueWrite with a value of Bits 0-3 other than as specified above, the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected).

#### 8.2.2.3 DP_SetVolume

- Datapoint type: DPT_Percent_U8 (5.004)
- Communication method: Group Object Number 3
- Flags: RWT

*Description*: The message comes from a new sound system device that can send a byte. The receiver sets the received volume value if it does not exceed the "Zone Max Volume" parameter. If the received value exceeds the "Zone Max Volume", yet within the permissible range, the set value is equal to "Zone Max Volume".

On receiving a groupValueWrite with a value outside the range [0..100] the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected).

**N.B.: It is used as an output to repeat the message to the Slave receivers as an alternative to the InfoVol datapoint. It is used as an output if the "PAR_AdjustableSelect" parameter is set to Vimar(=0) and the receiver is the Master Group.**

## SECTION 5: SOUND SYSTEM

#### 8.2.2.4 DP_SkipChannel

- Datapoint type: DPT_Step (1.007)
- Communication method: Group Object Number 4
- Flags: W

*Description*: The message comes from any device that can send an On/Off command. The receiver must perform the registration process on the next or previous valid channel (after checking the validity of the channel on the "Active Channels" configuration parameter) through the "Tx Generic Command" datapoint. If the procedure is successful the receiver tunes to this channel.

On receiving a groupValueWrite with a value outside the range [0..1] the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected).

#### 8.2.2.5 DP_SetChannel

- Datapoint type: DPT_Value_1_Ucount (5.010)
- Communication method: Group Object Number 5
- Flags: RWT

*Description*: The message comes from any device that can send a byte. The receiver must perform the registration process on the required channel (after checking the validity of the channel on the "Active Channels" configuration parameter) through the "Tx Generic Command" datapoint. If the procedure is successful it tunes to this channel. Valid values range between 0 and 3.

On receiving a groupValueWrite with a value outside the range [0..3] the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected).

**NB: it is used as an output to repeat the message to the Slave receivers as an alternative to the Ch Info datapoint. It is used as an output if the "PAR_AdjustableSelect" parameter is set to Vimar(=0) and the receiver is the Master Group.**

#### 8.2.2.6 DP_Ch+/Track+

- Datapoint type: Vimar Proprietors
- Communication method: Group Object Number 6
- Flags: W

*Format*: 1bit (1=Ch+, 0=Track+)

*Description*: The message comes from a rocker button for a By-me or sound system that can send an On/Off command. On pressing the top button an On command is sent that is equivalent to Ch+, on pressing the bottom button an Off command is sent that is equivalent to Track+. On receiving the Ch+ message the receiver must perform the registration process on the required channel (after checking the validity of the channel on the "Active Channels" configuration parameter) through the "Tx Generic Command" datapoint. If the procedure is successful it tunes to this channel. On receiving the Track+ message the receiver must communicate the request to skip track to Tx through the "Tx Generic Command" datapoint.
On receiving a groupValueWrite with a value outside the range [0..1] the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected).

#### 8.2.2.7 DP_Sleep

- Datapoint type: DPT_Enable (1.003)
- Communication method: Group Object Number 7
- Flags: WT

*Description*: The message comes from any device that can send an On or Off command. The receiver must switch on on the last channel in which it was tuned for a time determined by the "Sleep Time" configuration parameter. Upon receiving the OFF message the receiver must switch off. Each time the channel or the volume is changed, during Sleep time, or a new Sleep message arrives with ON, the Sleep time must be reloaded.

On receiving a groupValueWrite with a value outside the range [0..1] the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected).

**NB: it is used as an output to repeat the message to the Slave receivers as an alternative to the Sleep Info datapoint. It is used as an output if the "PAR_AdjustableSelect" parameter is set to Vimar(=0) and the receiver is the Master Group.**

#### 8.2.2.8 DP_Mute

- Datapoint type: DPT_Enable (1.003)
- Communication method: Group Object Number 9
- Flags: W

*Description*: The message comes from any device that can send an On or Off command. The receiver, if turned on, must set the volume equal to the "Volume Mute" parameter when On and return to the previous condition when Off. During the Mute time the receiver accepts no volume changes (SetVolume or DimmingVolume).

On receiving a groupValueWrite with a value outside the range [0..1] the device does nothing (as specified by KNX for groupValueWrite with a different data length than expected)

# SECTION 5: SOUND SYSTEM

## 9. Audio amplifier 10+10W on DIN rail

There is only one FB in the device.

| Fb Index | Fb Type |
|---|---|
| 1 | AudioRxDINAudioOutput |
| Audio output for controlling 2 10W speakers. | |

### 9.1 AudioRxDINAudioOutput

For a description of the communication datapoints please refer to the "AudioRx2MAudioOutput" section in the previous chapter "2M flush mounting 1+1W amplifier equipped with control buttons", since the functional blocks have the same model.

## 10. 4-button home automation ampl. control 1+1 W

The device consists of the following Functional Blocks:

| Fb Index | Fb Type |
|---|---|
| 1 | FBT_BUTTON (0x6C) |
| 2 | FBT_BUTTON (0x6C) |
| 3 | FBT_BUTTON (0x6C) |
| 4 | FBT_BUTTON (0x6C) |
| 5 | FBT_ROCKER BUTTON (0x6D) |
| 6 | FBT_ROCKER BUTTON (0x6D) |
| 7 | FBT_OUT_AUDIO (0x6F) |

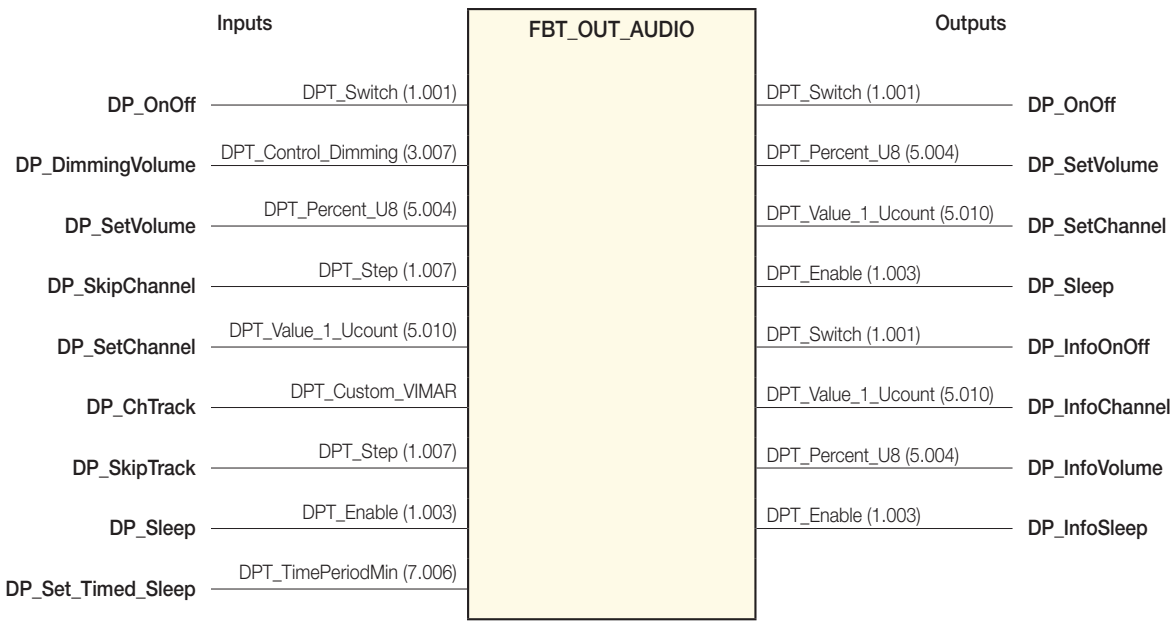
### 10.1 FBT_BUTTON

For the description of the communication datapoints refer to the Functional block paragraph “VIMAR 3.0 single push buttons” (FBT_BUTTON) in the Light and roller shutter control section.

### 10.2 FBT_ROCKER BUTTON

For the description of the communication datapoints refer to the Functional block paragraph “VIMAR 3.0 rocker buttons” (FBT_ROCKER BUTTON) in the Light and roller shutter control section.

### 10.3 FBT_OUT_AUDIO



## SECTION 5: SOUND SYSTEM

| Name in VIEW Pro | Datapoint | DPT | CO number | Flags |
|---|---|---|---|---|
| **DPTx_AudioOnOff** | **DP_OnOff** | DPT_Switch (1.001) | 1 | RWT |
| Message received as a request to switch on the zone, while it is used at the output to repeat the message to the slaves. | | | | |
| **DPTx_VolumeControl** | **DP_DimmingVolume** | DPT_Control_Dimming (3.007) | 2 | W |
| Message received as a request to increase or decrease the receiver volume. | | | | |
| **DPTx_VolumeValue** | **DP_SetVolume** | DPT_Percent_U8 (5.004) | 3 | RWT |
| Message received as a request to set the volume, while it is used to repeat the volume value to the slave receivers. | | | | |
| **DPTx_AudioChSkipWithdrawn** | **DP_SkipChannel** | DPT_Step (1.007) | 4 | W |
| Message received as a request to change the listening channel. | | | | |
| **DPTx_Channel** | **DP_SetChannel** | DPT_Value_1_Ucount (5.010) | 5 | RWT |
| Message received as a request to set the channel, while it is used to repeat the channel value to the slave receivers. | | | | |
| **DPTx_SkipChannelTrack** | **DP_ChTrack** | DPT_Custom_VIMAR | 6 | W |
| Message received as a request to increase the channel and increase the track. | | | | |
| **DPTx_SkipTrack** | **DP_SkipTrack** | DPT_Step (1.007) | 7 | W |
| Message received as a request to go to the next track. | | | | |
| **DPTx_Sleep** | **DP_Sleep** | DPT_Enable (1.003) | 8 | RWT |
| Message received as a sleep request, i.e. for switching on for a time equal to the Sleep Time parameter, while it is used to repeat the sleep request to the slave receivers. | | | | |
| **DPTx_SetTimedSleep** | **DP_Set_Timed_Sleep** | DPT_TimePeriodMin (7.006) | 13 | W |
| Message received as a request to activate the Sleep function for a time equal to the time contained in the message. | | | | |
| **DPTx_AudioOnOffInfo** | **DP_InfoOnOff** | DPT_Switch (1.001) | 15 | RT |
| Message used to relay information on the receiver status to the slave receivers | | | | |
| **DPTx_InfoChannel** | **DP_InfoChannel** | DPT_Value_1_Ucount (5.010) | 16 | RT |
| Message used to relay information on the current channel to the slave receivers | | | | |
| **DPTx_InfoVolume** | **DP_InfoVolume** | DPT_Percent_U8 (5.004) | 17 | RT |
| Message used to relay information on the current volume to the slave receivers | | | | |
| **DPTx_InfoSleep** | **DP_InfoSleep** | DPT_Enable (1.003) | 18 | RT |
| Message used to relay the on or off request to the slave receivers | | | | |

## SECTION 5: SOUND SYSTEM

### 11. Home automation system control with 4 push buttons +1 2M LINEOUT

The device consists of the following Functional Blocks:

| Fb Index | Fb Type |
|---|---|
| 1 | FBT_BUTTON (0x6C) |
| 2 | FBT_BUTTON (0x6C) |
| 3 | FBT_BUTTON (0x6C) |
| 4 | FBT_BUTTON (0x6C) |
| 5 | FBT_ROCKER BUTTON (0x6D) |
| 6 | FBT_ROCKER BUTTON (0x6D) |
| 7 | FBT_OUT_AUDIO (0x6F) |

#### 10.1 FBT_BUTTON

For the description of the communication datapoints refer to the Functional block paragraph “VIMAR 3.0 single push buttons” (FBT_BUTTON) in the Light and roller shutter control section.

#### 10.2 FBT_ROCKER BUTTON

For the description of the communication datapoints refer to the Functional block paragraph “VIMAR 3.0 rocker buttons” (FBT_ROCKER BUTTON) in the Light and roller shutter control section.

#### 10.3 FBT_OUT_AUDIO

For the description of the communication datapoints refer to paragraph FBT_OUT_AUDIO in the previous chapter.

CE

By-me Plus KNX EN 02 2107

VIMAR
Viale Vicenza, 14
36063 Marostica VI - Italy
www.vimar.com