KNX Rain sensor

GW 90881

Technical Manual

Contents

1 Introduction

This manual explains the functions of the **“KNX rain sensor”** (GW90881), and how they are set and configured with the aid of the ETS configuration software.

2 Application

This sensor detects wet or dry atmospheric conditions and then implements the appropriate commands.
The rain sensor is also equipped with four AND logic functions and four OR logic functions with four inputs.

2.1 Association limits

The maximum number of communication objects available is 34.
The maximum number of associations that the device can store is 255.
The maximum number of group addresses is 254.

3 *"General Settings "* menu

The ***General settings*** menu contains the parameters used to configure general device behaviour. Fig. 3.1 shows the default screen of the ***General settings*** menu, with the relative settable parameters.

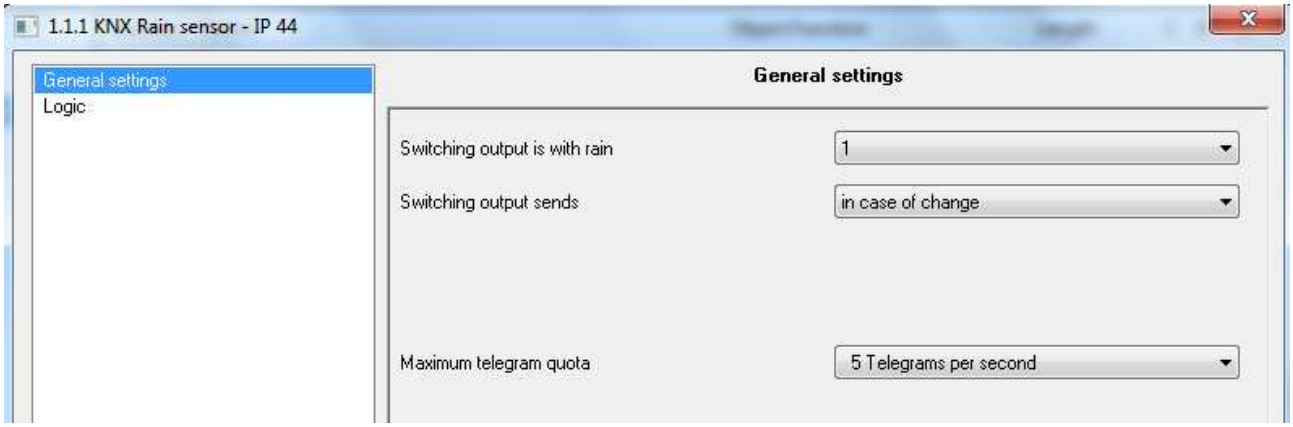

Fig. 3.1

3.1 Parameters

➢ *3.1.1 Switching object value when rain is detected (=Switching output rain)*

This parameter is used to set the value to be sent via the 0 **Rain switching** communication object, when rain is detected. The possible values are **0** or **1**.

➢ *3.1.2 The output object is sent*

This parameter is used to define when the 0 **Rain switching** communication object must be sent. The possible values are:

- ***never***
  in this way, the 0 **Rain switching** object is never sent.

- ***with variations***
  This is the default setting. The 0 **Rain switching** object is sent when there is a change from wet to dry conditions or vice versa; the change from wet to dry is detected after about 8 minutes without rain.

- ***with variation 0 -> 1***
  The 0 **Rain switching** object is only sent when the 0 status changes to 1, according to how the value of the "switching with rain detected" object has been configured (see paragraph 3.1.1);
  the change from wet to dry is detected after about 8 minutes without rain.

- ***with variation 1 -> 0***
  The 0 **Rain switching** object is only sent when the 1 status changes to 0, according to how the value of the "switching with rain detected" object has been configured (see paragraph 3.1.1);
  the change from wet to dry is detected after about 8 minutes without rain.

- ***with variation, and cyclically***
  The 0 **Rain switching** object is sent when the wet/dry status changes and cyclically (with a frequency configured in the **Cyclical sending period** parameter); the change from wet to dry is detected after about 8 minutes without rain.

- ***with variation 0 -> 1 and cyclically***
  The 0 **Rain switching** object is sent when the 0 status changes to 1, according to how the value of the "switching with rain detected" object has been configured (see paragraph 3.1.1), and cyclically (with a frequency configured in the **Cyclical sending period** parameter);
  the change from wet to dry is detected after about 8 minutes without rain.

- ***with variation 1 -> 0 and cyclically***
  The 0 **Rain switching** object is sent when the 1 status changes to 0, according to how the value of the "switching with rain detected" object has been configured (see paragraph 3.1.1), and cyclically (with a frequency configured in the **Cyclical sending period** parameter);
  the change from wet to dry is detected after about 8 minutes without rain.

➢ *3.1.3 Maximum telegram transmission speed (=Maximum telegram quota)*

This parameter determines the maximum telegram transmission speed in one second. The settable values vary from 1 telegram per second, to 20.
The default value is 5.

3.2 Communication objects

The communication objects enabled from the ***General settings*** menu are shown below (see fig. 3.2).

Number	Name	Object Function	Length	C	R	W	T	U	Data Type
0	Switching output rain	Output	1 bit	C	R	-	T	-	1 bit DPT_Switch
33	Software Version	Readable	2 Byte	C	R	-	-	-	2 byte unsigned value

Fig 3.2

➢ *3.2.1 Rain switching*

This communication object transmits a value (1 or 0, depending on the settings defined by the relative parameter to detect the wet or dry status) on the BUS.
Rain is indicated immediately. The change from wet to dry conditions is detected after about 8 minutes without rain (the time needed for the heater to evaporate the water).
The enabled flags are C (communication), R (reading from BUS), T (transmission).
The standardised object format is *1.001 DPT_Switch*, the dimension of the object is *1 bit*, and the commands that it sends are *dry / rain*.

➢ *3.2.2 Software version*

This communication object is always active. Via a reading request, it indicates the software version of the sensor.
The enabled flags are C (communication) and R (reading from BUS).
The standardised object format is 217*.001 DPT_Version*, the dimension of the object is 2 *Bytes* , and it receives a reading request to send the *software version* ID.

4 *"Logic"* menu

The ***Logic*** menu contains the parameters for enabling the communication objects for the logic inputs, and those for configuring logic functions of the AND/OR type; with regards the logics, there are four blocks for both AND logic and OR logic, as shown in figure 4.1.

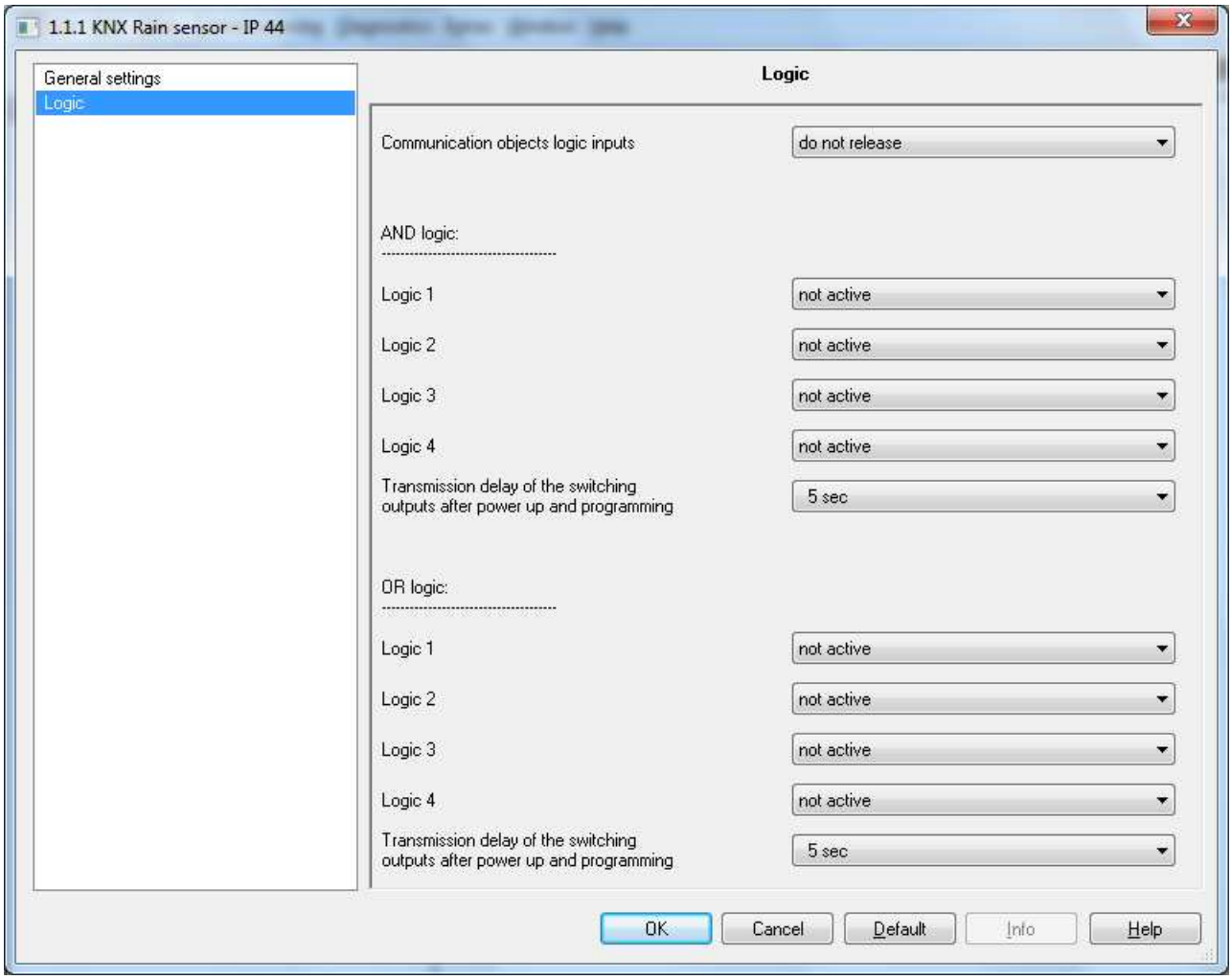

Fig. 4.1

4.1 Parameters

➢ *4.1.1 Communication objects for logic inputs*

With this setting, you can enable the communication objects inherent to the logic inputs, for use with the logic functions.

- ***Disable***
  The logic inputs are not visible.

- ***Enable***
  This enables the 8 **Logic input** communication objects from 25 to 32.

➢ *4.1.2 AND logics from 1 to 4*

By activating a logic block, you can make a series of parameters available (see figure 4.2) for **AND logic 1**.

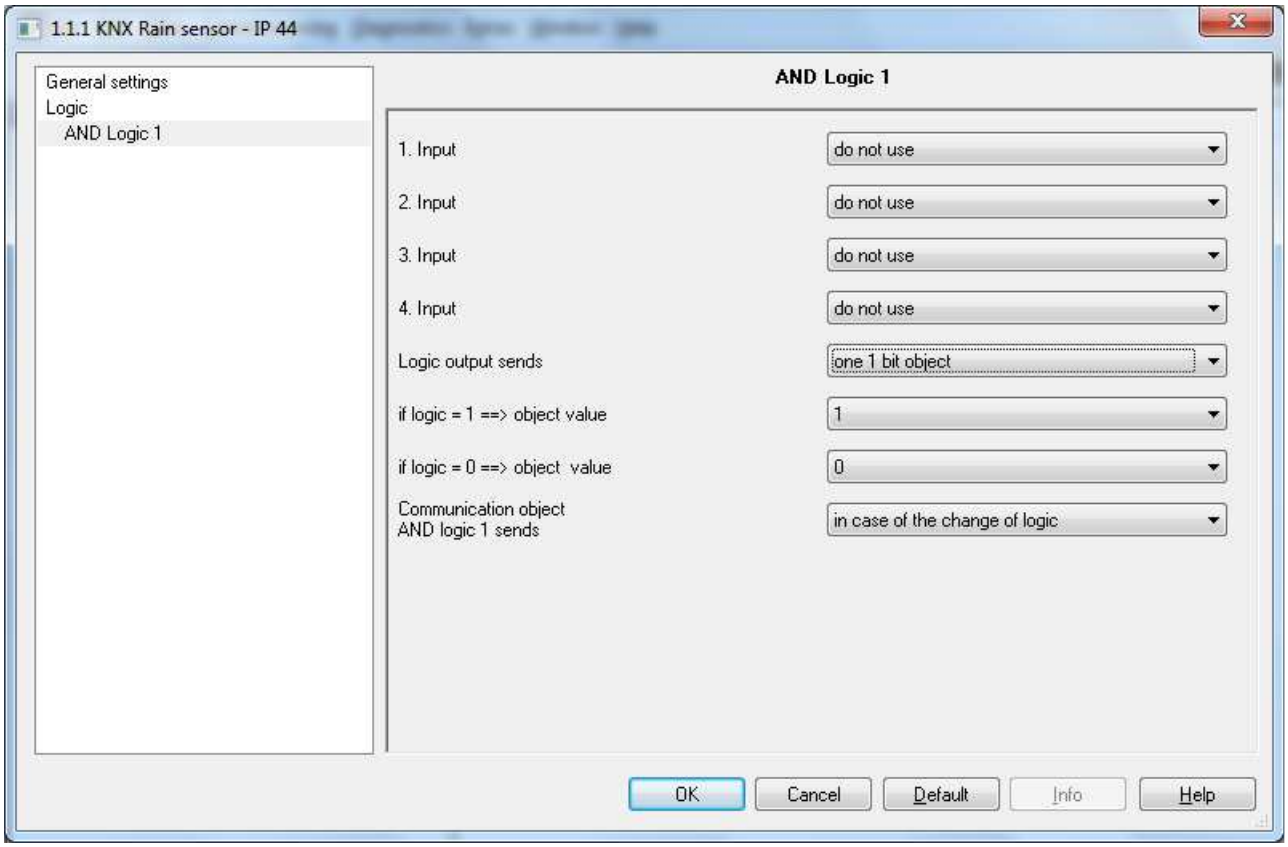

Fig. 4.2

Each logic block allows you to create a function with up to four inputs.
For each of the four inputs, you can directly associate an event relating to the rain sensor plus a *logic input communication object*; in the latter case, the value of the logic input used is defined by a group address in ETS which includes the logic input object used (to enable it, refer to paragraph 4.1.1), and another communication object (e.g. the switching of an input interface).

The **result of the logic operation sends** parameter activates one 1-bit object or two 8-bit objects:

select **one 1-bit object** to activate the **If logic =1 ==> object value** and **If logic =0 ==> object value** parameters, as shown in figure 4.2. In the case of the first logic block, when the logic is true (i.e.=1), these two parameters define whether object 1 **AND logic 1** must send a value of 1 or 0; in the same way, when the logic is false (i.e. =0), they define the value to be sent for object 1 **AND logic 1.**
The **communication object AND logic 1 sends** parameter enables the sending of the communication object, on the basis of the following options:

- **Only if the logic outcome changes**
  This means the object is sent when the value of the logic changes
- **Only if the logic outcome is 1**
  this means the object is sent when the value of the logic is 1
- **Only if the logic outcome is 0**
  this means the object is sent when the value of the logic is 0
- **If the logic outcome changes, and cyclical sending**

This means the object is sent when the value of the logic changes, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).

- **If the logic outcome is 1, and cyclical sending**
  This means the object is sent when the value of the logic is 1, and cyclically (with a frequency defined in the "Cyclical sending period" parameter).

- **If the logic outcome is 0, and cyclical sending**
  This means the object is sent when the value of the logic is 0, and cyclically (with a frequency defined in the "Cyclical sending period" parameter).

Select **two 8-bit objects** to activate the parameters shown in figure 4.3. These four parameters are used to send two values, both when the logic is true and when it is false. The possible values range from 0 to 255.
The objects activated for the first logic block are object 2 **AND logic 1** with the *8-bit A output* function, and object 3 **AND logic 1** with the *8-bit B output* function.

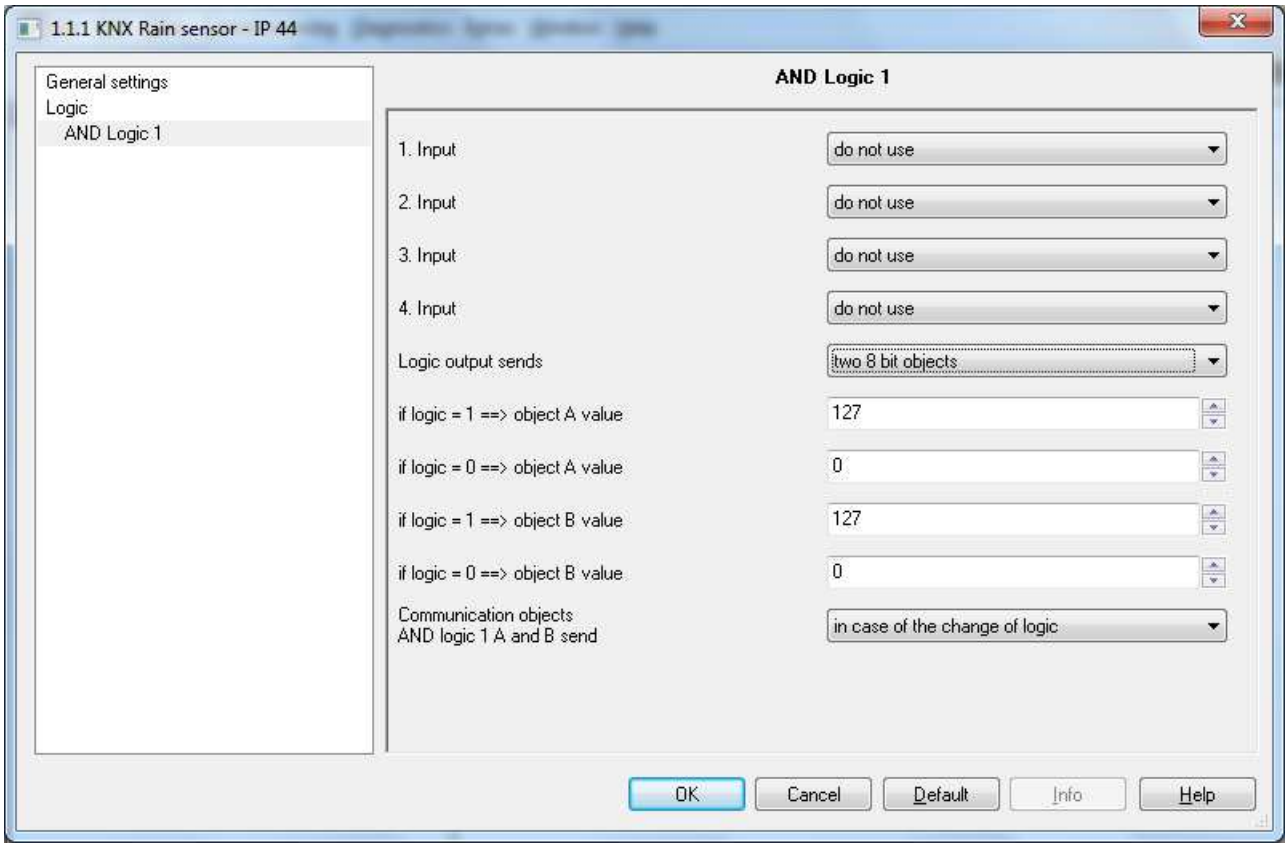

Fig. 4.3

The **communication objects AND logic 1 A and B send** parameter enables the sending of the communication objects on the basis of the following options:

- **Only if the logic outcome changes**
  This means the object is sent when the value of the logic changes

- **Only if the logic outcome is 1**
  This means the object is sent when the value of the logic is 1

- **Only if the logic outcome is 0**
  This means the object is sent when the value of the logic is 0

- **If the logic outcome changes, and cyclical sending**
  This means the object is sent when the value of the logic changes, and cyclically (with a frequency defined in the "Cyclical sending period" parameter).

- **If the logic outcome is 1, and cyclical sending**
  This means the object is sent when the value of the logic is 1, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).

- **If the logic outcome is 0, and cyclical sending**
  This means the object is sent when the value of the logic is 0, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).

➢ *4.1.4 OR logics from 1 to 4*

By activating a logic block, you can make a series of parameters available (see figure 4.4) for **OR logic 1**.

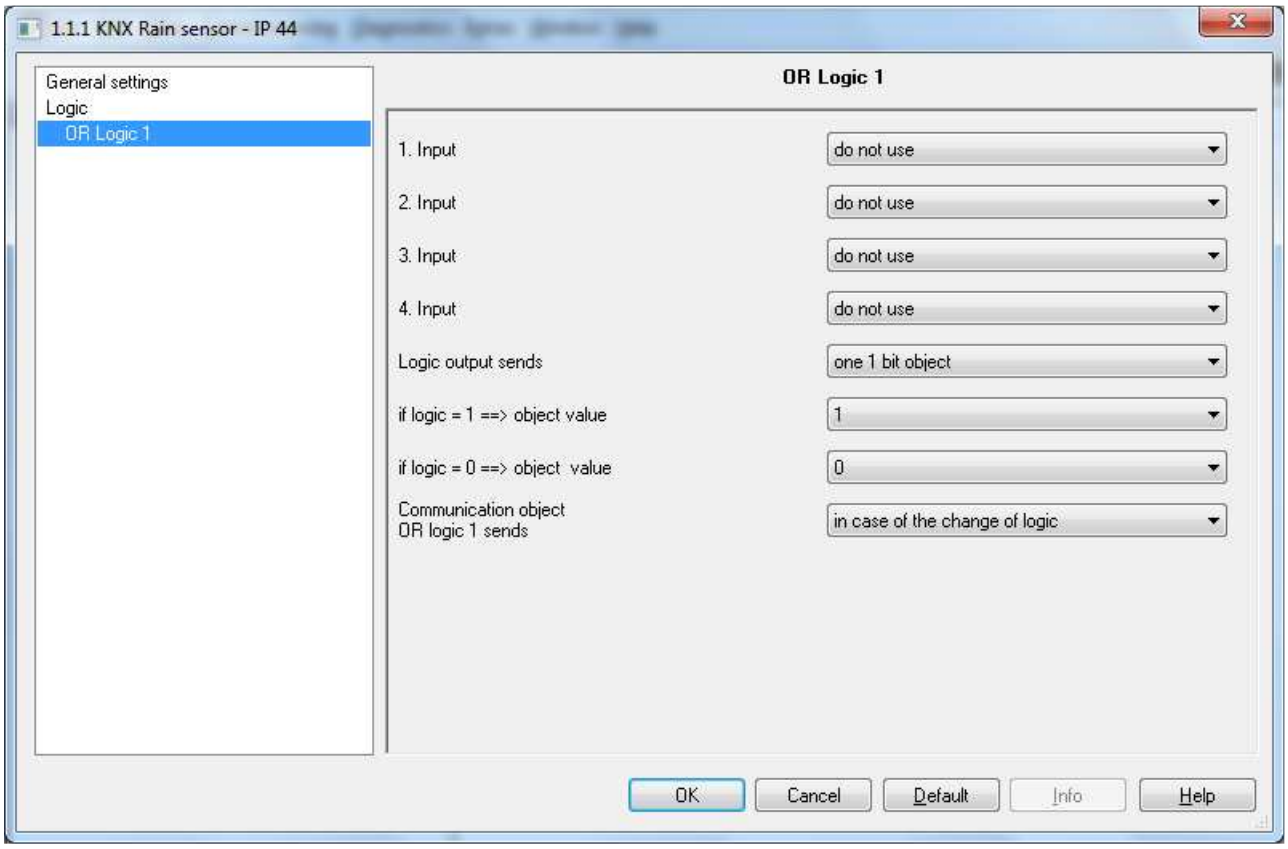

Fig. 4.4

Each logic block allows you to create a function with up to four inputs.
For each of the four inputs, you can directly associate an event relating to the rain sensor plus a *logic input communication object*; in the latter case, the value of the logic input used is defined by a group address in ETS which includes the logic input object used (to enable it, refer to paragraph 4.1.1), and another communication object (e.g. the switching of an input interface).

The **result of the logic operation sends** parameter activates one 1-bit object or two 8-bit objects:

select **one 1-bit object** to activate the **If logic =1 ==> object value** and **If logic =0 ==> object value** parameters, as shown in figure 4.4. In the case of the first logic block, when the logic is true (i.e.=1), these two parameters define whether object 13 **OR logic 1** must send a value of 1 or 0; in the same way, when the logic is false (i.e. =0), they define the value to be sent for object 1 **OR logic 1.**
The **communication object OR logic 1 sends** parameter enables the sending of the communication object, on the basis of the following options:

- **Only if the logic outcome changes**
  This means the object is sent when the value of the logic changes

- **Only if the logic outcome is 1**
  This means the object is sent when the value of the logic is 1

- **Only if the logic outcome is 0**
  This means the object is sent when the value of the logic is 0

- **If the logic outcome changes, and cyclical sending**
  This means the object is sent when the value of the logic changes, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).

- **If the logic outcome is 1, and cyclical sending**
  This means the object is sent when the value of the logic is 1, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).

- **If the logic outcome is 0, and cyclical sending**
  This means the object is sent when the value of the logic is 0, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).

Select **two 8-bit objects** to activate the parameters shown in figure 4.5. These four parameters are used to send two values, both when the logic is true and when it is false. The possible values range from 0 to 255.
The objects activated for the first logic block are object 14 **OR logic 1** with the *8-bit A output* function, and object 15 **OR logic 1** with the *8-bit B output* function.

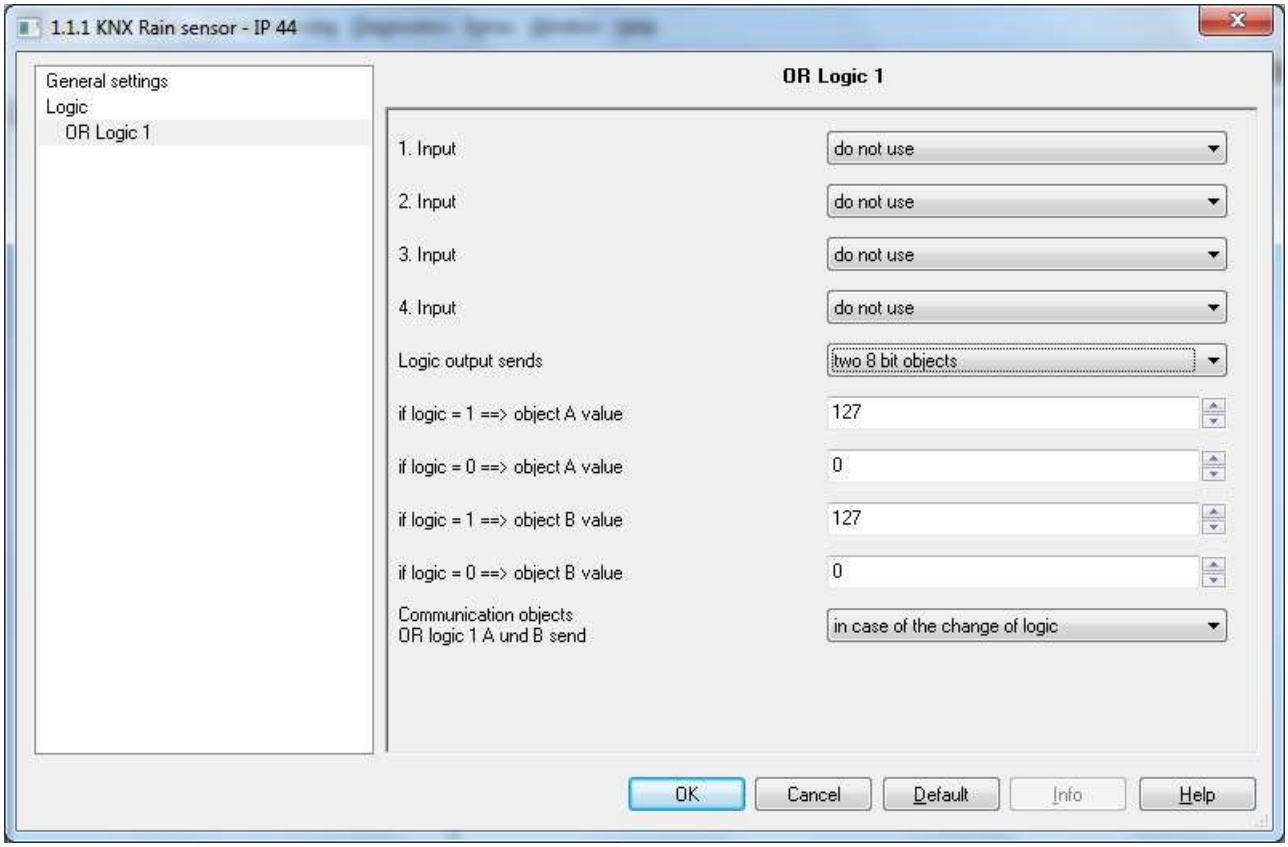

Fig. 4.5

The **communication objects OR logic 1 A and B send** parameter enables the sending of the communication objects on the basis of the following options:

- **Only if the logic outcome changes**
  This means the object is sent when the value of the logic changes
- **Only if the logic outcome is 1**
  This means the object is sent when the value of the logic is 1
- **Only if the logic outcome is 0**
  This means the object is sent when the value of the logic is 0
- **If the logic outcome changes, and cyclical sending**
  This means the object is sent when the value of the logic changes, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).
- **If the logic outcome is 1, and cyclical sending**
  This means the object is sent when the value of the logic is 1, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).
- **If the logic outcome is 0, and cyclical sending**
  This means the object is sent when the value of the logic is 0, and cyclically (with a frequency defined in the "Cyclic sending period" parameter).

➢ *4.1.5 Delayed sending of output objects on download and BUS voltage recovery*

This parameter is available for both the AND and OR logic functions. It defines the delay time with which the objects are sent with output function, both on the download of the application and on BUS voltage recovery.

4.2 Communication objects

The communication objects enabled from the ***Logic*** menu and subsequent menus are shown below (see fig. 4.6).

Number	Name	Object Function	Length	C	R	W	T	U	Data Type
0	Switching output rain	Output	1 bit	C	R	-	T	-	1 bit DPT_Switch
1	AND logic 1	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
4	AND logic 2	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
7	AND logic 3	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
10	AND logic 4	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
13	OR logic 1	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
16	OR logic 2	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
19	OR logic 3	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
22	OR logic 4	Switching output	1 bit	C	R	-	T	-	1 bit DPT_Bool
25	Logic input 1		1 bit	C	R	W	-	-	1 bit DPT_Bool
26	Logic input 2		1 bit	C	R	W	-	-	1 bit DPT_Bool
27	Logic input 3		1 bit	C	R	W	-	-	1 bit DPT_Bool
28	Logic input 4		1 bit	C	R	W	-	-	1 bit DPT_Bool
29	Logic input 5		1 bit	C	R	W	-	-	1 bit DPT_Bool
30	Logic input 6		1 bit	C	R	W	-	-	1 bit DPT_Bool
31	Logic input 7		1 bit	C	R	W	-	-	1 bit DPT_Bool
32	Logic input 8		1 bit	C	R	W	-	-	1 bit DPT_Bool
2	AND logic 1	8 bit Output A	1 Byte	C	R	-	T	-	
3	AND logic 1	8 bit Output B	1 Byte	C	R	-	T	-	
5	AND logic 2	8 bit Output A	1 Byte	C	R	-	T	-	
6	AND logic 2	8 bit Output B	1 Byte	C	R	-	T	-	
8	AND logic 3	8 bit Output A	1 Byte	C	R	-	T	-	
9	AND logic 3	8 bit Output B	1 Byte	C	R	-	T	-	
11	AND logic 4	8 bit Output A	1 Byte	C	R	-	T	-	
12	AND logic 4	8 bit Output B	1 Byte	C	R	-	T	-	
14	OR logic 1	8 bit Output A	1 Byte	C	R	-	T	-	
15	OR logic 1	8 bit Output B	1 Byte	C	R	-	T	-	
17	OR logic 2	8 bit Output A	1 Byte	C	R	-	T	-	
18	OR logic 2	8 bit Output B	1 Byte	C	R	-	T	-	
20	OR logic 3	8 bit Output A	1 Byte	C	R	-	T	-	
21	OR logic 3	8 bit Output B	1 Byte	C	R	-	T	-	
23	OR logic 4	8 bit Output A	1 Byte	C	R	-	T	-	
24	OR logic 4	8 bit Output B	1 Byte	C	R	-	T	-	

Fig. 4.6

➢ *4.2.1 AND logic 1 to 8 - 1-bit output function*

Eight 1-bit communication objects are available, one for each AND logic output. They are used to send the logic value, on the basis of the established parameters.
The objects in question are 1-4-7-10, and the enabled flags are C (communication), R (reading from BUS) and T (transmission).
The standardised object format is *1.002 DPT_Bool*, so the object dimension is 1 *bit*. It assumes the logic values of *zero/ one*.

➢ *4.2.2 AND logic 1 to 8 - 8-bit x output function*

As an alternative to the 1-bit output, there are two 8-bit objects for the output of each AND logic. The *x* in the function specifies whether it is object A or B.

On the basis of the established parameters, these 8-bit communication objects are used to send two values (between 0 and 255) when the logic is true, and two values (between 0 and 255) when the logic is false.
The objects in question are 2-3 for the AND 1 logic, 5-6 for the AND 2 logic, 8-9 for the AND 3 logic, and 11-12 for the AND 4 logic.
The enabled flags are C (communication), R (reading from BUS), T (transmission).
The standardised object format is 5*.010 DPT_1byte_unsigned_value*, so the object dimension is 8 *bit* and it assumes values between 0 and 255.

➢ *4.2.3 OR logic 1 to 8 - 1-bit output function*

Eight 1-bit communication objects are available, one for each OR logic output. They are used to send the logic value, on the basis of the established parameters.
The objects in question are 13-16-19-22, and the enabled flags are C (communication), R (reading from BUS) and T (transmission).
The standardised object format is *1.002 DPT_Bool*, so the object dimension is 1 *bit*. It assumes the logic values of *zero/ one*.

➢ *4.2.4 OR logic 1 to 8 - 8-bit x output function*

As an alternative to the 1-bit output, there are two 8-bit objects for the output of each OR logic. The *x* in the function specifies whether it is object A or B.
On the basis of the established parameters, these 8-bit communication objects are used to send two values (between 0 and 255) when the logic is true, and two values (between 0 and 255) when the logic is false.
The objects in question are 14-15 for the OR 1 logic, 17-18 for the OR 2 logic, 20-21 for the OR 3 logic, and 23-24 for the OR 4 logic.
The enabled flags are C (communication), R (reading from BUS), T (transmission).
The standardised object format is 5*.010 DPT_1byte_unsigned_value*, so the object dimension is 8 *bit* and it assumes values between 0 and 255.

➢ *4.2.5 Logic 1 .. 8 input*

Eight communication objects (from 25 to 32) are available. They can receive BUS telegrams whose value (or inverted signal) forms the input of a logic function.
The enabled flags are C (communication), R (reading from BUS), and W (writing from BUS).
The standardised object format is *1.002 DPT_Bool*, so the object dimension is 1 *bit*. It assumes the logic values of *zero/ one*.