

**Movement detector with EIB twilight sensor**



**GW 10 786**  
**GW 12 786**  
**GW 14 786**

**TECHNICAL MANUAL**

## Summary

1	Introduction.....	3
2	Application .....	4
2.1	Limits to the associations .....	4
3	“Times” Menu .....	5
3.1	Parameters.....	6
3.2	Communication objects.....	7
4	“Twilight sensor” menu .....	8
4.1	Parameters.....	8
4.2	Communication objects.....	9
5	“Movement sensor” menu .....	10
5.1	Parameters.....	10
5.2	Communication objects.....	13
6	“Block” menu .....	15
6.1	Parameters.....	15
6.2	Communication objects.....	16
7	“Aux x” Menu .....	17
7.1	Parameters.....	17
7.2	Communication objects.....	19
8	“AuxX_Block” menu.....	20
8.1	Parameters.....	20
8.2	Communication objects.....	21

# 1 Introduction

This manual describes the functions of the devices named GW10786 - GW12786 - GW14786 "**Movement Detector with EIB twilight sensor**" and how to use the ETS configuration software to change the settings and configurations.

## 2 Application

The Movement Detector with EIB Twilight Sensor is a device that is able to send bus commands to other devices on the Building Automation Konnex system according to the movements detected and the brightness measured by the two incorporated sensors; this device is in fact able to send ON/OFF commands, percentage values and scene activation commands to other KNX/EIB devices following movement detection, whether conditioned or not by the level of light in the environment where the device is installed.

The device is designed to also operate as a simple twilight sensor.

Three auxiliary command objects are also available which can be used independently to activate or command other devices according to the conditions which may occur.

It is also possible to enable/disable the function that sends commands following movement detection, including the commands sent by the auxiliary objects.

The device is fitted with two trimmers, one to set the brightness threshold and the other to modify the recovery time (that will be analysed in detail further on) and a pilot light backlit by a green led which indicates when a start and movement end telegram is sent.

This manual refers solely to the configuration using the ETS software. Please refer to the INSTALLATION AND USER MANUAL supplied with the product for instructions on how to adjust and use the trimmers fitted on the device.

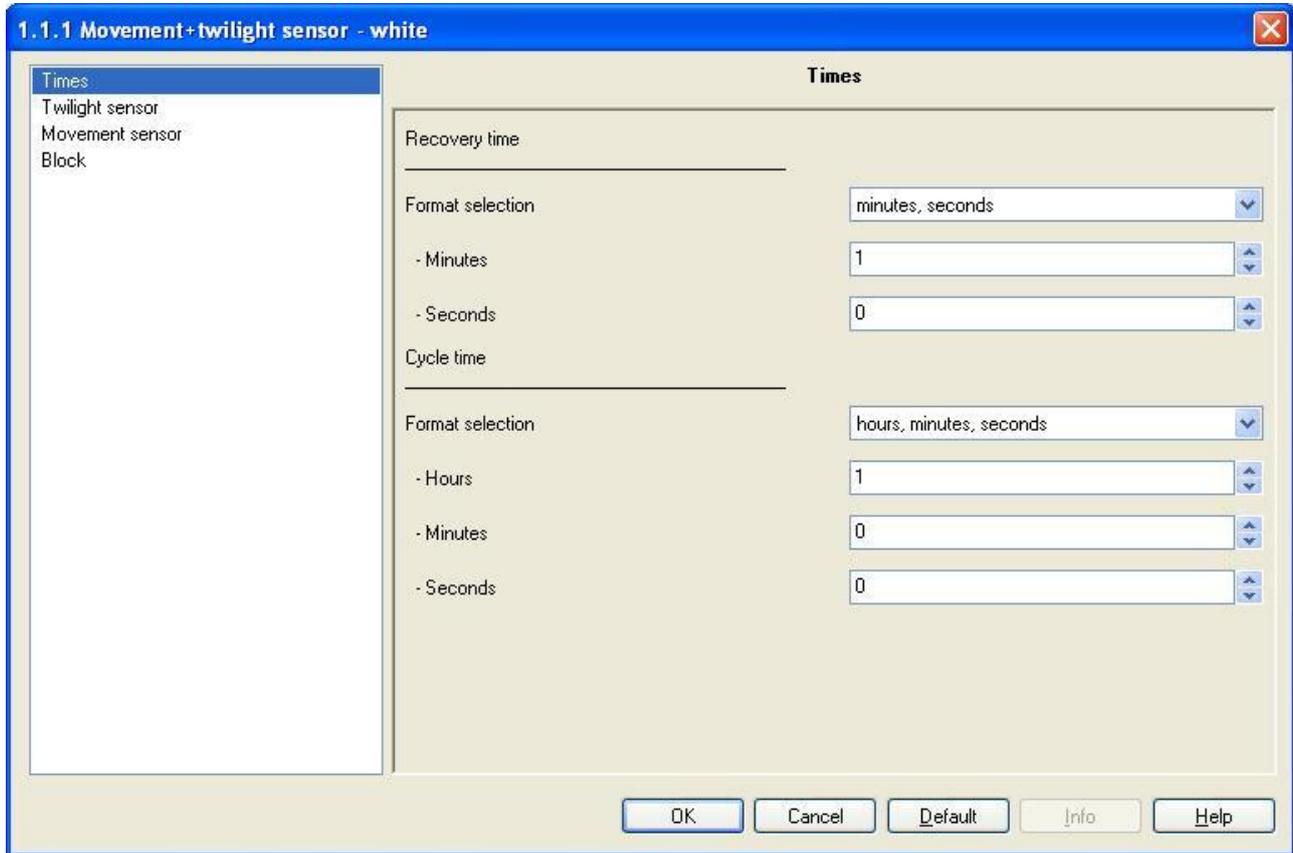
### 2.1 *Limits to the associations*

The maximum number of logical associations that the device is able to memorize is 40; this means that the maximum number of logical connections between communication objects and group addresses is 40.

The maximum number of group addresses that the device is able to memorize is 40; this means that it is possible to associate the communication objects to a maximum of 40 group addresses.

### 3 “Times” Menu

The *Times* menu lists all the parameters needed to configure the 1 bit inputs on device; Diag. 3.1 shows the complete *Times* menu screen shot with the relative configurable parameters.



Diag. 3.1

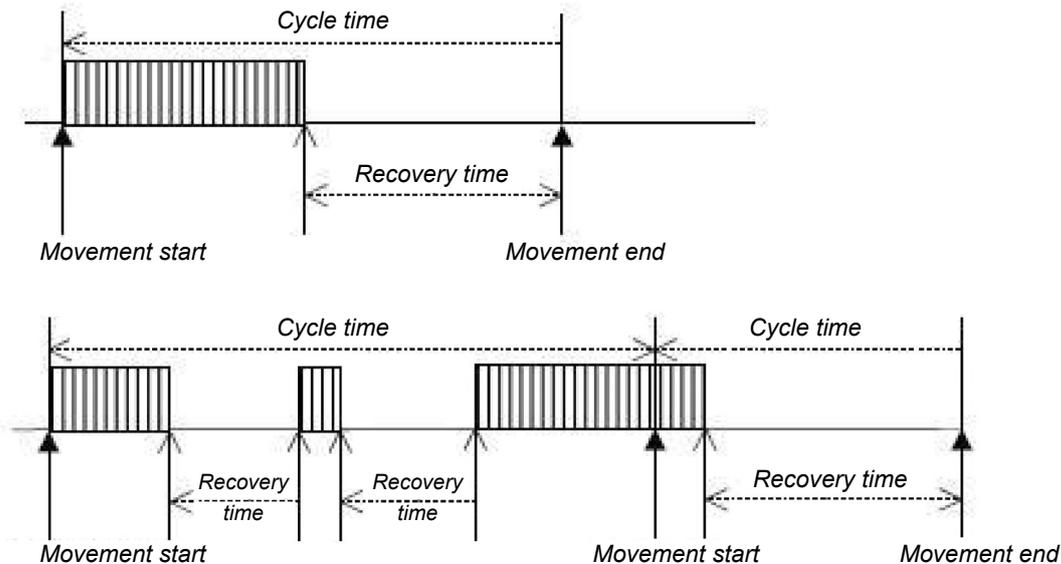
Before commencing a detailed analysis of the items on the configuration menu, let's look briefly at how the device actually functions so that the explanations of the items will then be easier to understand.

When the device detects movement (and after assessing the brightness level detected by the twilight sensor, if this option is enabled) it sends a movement start telegram; from this moment in time, the value on the cycle time counter and the recovery time counter both decrease.

The recovery time refers to the period that must lapse before the device can detect another movement within its range of action, so that it can determine the end of the movement and sent the relative “movement end” telegram (if this option is enabled); naturally, the recovery time counter is reset each time the device detects movement.

During the cycle time, any movement detected will not trigger the sending of a bus telegram as this would create an overflow on the bus line, whilst it is possible to set the repeated sending of periodical telegrams associated to the “movement start” event at the end of each cycle time; therefore, if the cycle time lapses and the recovery time has not expired, (that is, there is constant movement), the device will send another “movement start” telegram if the cyclic command mode is enabled.

Diag. 3.2 shows a drawing that may simply these concepts.



Diag. 3.2

The first diagram illustrates the condition where, within the cycle time, a time equal to the recovery time lapses without the device detecting any movement; the cycle time therefore commences when the first movement is detected, indicated by a bold font arrow; at the same time a movement start telegram is also sent (if the send telegram option is enabled); if no further movement is detected within the cycle time, the recovery time counter is reset, indicated by a styled font arrow. In this case, the recovery time count is completed because no movement has reset the counter value, and the end of the movement is confirmed by the sending of a movement end telegram (if this option is enabled) and the end of the cycle time count. Once the recovery time has expired, the detection of movement triggers the above illustrated cycle once again.

The second diagram replicates the complete operating condition, that is when more than one cycle time lapses before the movement end command is given.

When the device detects the first movement, indicated by a bold text arrow, this triggers the start of the cycle time counter and the sending of the movement start telegram (if this telegram is enabled); when no more movement is detected, the recovery time counter starts which is not terminated when a new movement is detected. This situation reoccurs again until the cycle time expires (second bold arrow); at this point, if the cyclic command option is enabled, the movement start telegram is sent once again and the cycle time counter is reset. This time however, a time equal to the recovery time lapses, within the cycle time, without any movement detection and this condition triggers the movement end command (last bold arrow), and the movement end telegram is sent, and the cycle time count ends. The detection of movement triggers the above illustrated cycle once again.

## 3.1 Parameters

### ➤ 3.1.1 Format selection

This determines the format to be set for the recovery time.

According to the value set for this item, the **Minutes** and **seconds** options will be displayed.

The settings are:

- **seconds**

On using this value the recovery time format will be in seconds, so the maximum value will be 59 seconds. With this setting, the **Seconds** option is available.

- **minutes, seconds**

On using this value the recovery time format will be in minutes and seconds, so the maximum value will be 59 minutes and 59 seconds and the minimum value 1 minute. With this setting, the **Minutes** and **Seconds** options is available.

### ➤ 3.1.2 Minutes

Here you can set the first of the two values (minutes) for the recovery time; the values range from 1 (minute) to 59 (minutes).

### ➤ 3.1.3 Seconds

Here you can set the second of the two values (seconds) for the recovery time; the values range from 10 (seconds) to 59 (minutes) in the case where the **Format selection** option is set to **seconds**, otherwise the values will range from 0 (seconds) to 59 (seconds).

### ➤ 3.1.4 Format selection

This determines the format to be set for the cycle time.

According to the value set for this item, the **Hours**, **Minutes** and **seconds** options will be displayed.

The settings are:

- **seconds**

On using this value the cycle time format will be in seconds, so the maximum value will be 59 seconds. With this setting, the **Seconds** option is available.

- **minutes, seconds**

On using this value the cycle time format will be in minutes and seconds, so the maximum value will be 59 minutes and 59 seconds and the minimum value 1 minute. With this setting, the **Minutes** and **Seconds** options is available.

- **hours, minutes, seconds**

On using this value the cycle time format will be in hours, minutes and seconds, so the maximum value will be 59 hours and 59 minutes and 59 seconds and the minimum value 1 hour. With this setting, the **Hours**, **Minutes** and **Seconds** options is available.

### ➤ 3.1.5 Hours

Here you can set the first of the three values (hours) for the cycle time; the values range from 1 (hour) to 59 (hours).

### ➤ 3.1.6 Minutes

Here you can set the second of the three values (minutes) for the cycle time; the values range from 1 (minute) to 59 (minutes) in the case where the **Format selection** option is set to **minutes, seconds**, otherwise the values will range from 0 (minutes) to 59 (minutes).

### ➤ 3.1.7 Seconds

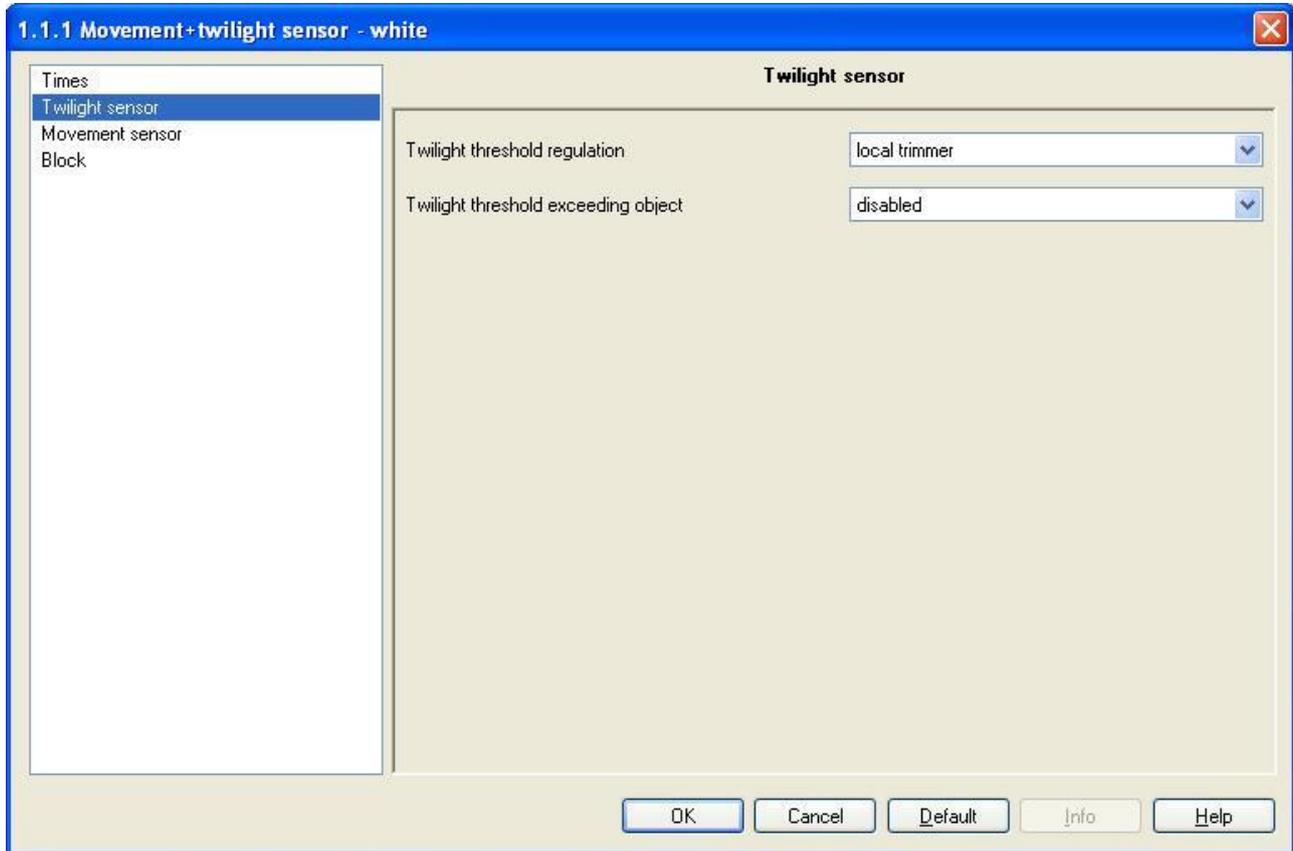
Here you can set the third of the three values (seconds) for the cycle time; the values range from 20 (seconds) to 59 (minutes) in the case where the **Format selection** option is set to **seconds**, otherwise the values will range from 0 (seconds) to 59 (seconds).

## 3.2 Communication objects

There are no communication objects enabled by the **Times** menu.

## 4 “Twilight sensor” menu

The **Twilight sensor** menu lists all the parameters needed to configure the function of the twilight sensor on the device; Diag. 4.1 shows the complete **Twilight sensor** menu screen shot with the relative configurable parameters.



Diag. 4.1

### 4.1 Parameters

#### ➤ 4.1.1 Twilight threshold regulation

Here you can configure the brightness value that determines the threshold for the twilight sensor. The settings are:

- **local trimmer**

The regulation of the brightness value that constitutes the twilight sensor threshold is performed by a local trimmer installed on the device. With this setting the **Threshold value** option is not visible..

- **parameter**

The regulation of the brightness value that constitutes the twilight sensor threshold is performed by an ETS parameter and it cannot be modified by the local trimmer installed on the device. With this setting, the **Threshold value** option is available.

#### ➤ 4.1.2 Threshold value

Here you can set the brightness value that constitutes the threshold for the twilight sensor. The values range from 10 lux to 500 lux; the threshold value in this case cannot be modified by the local trimmer installed on the device.

### ➤ 4.1.3 *Twilight threshold exceeding object*

This section allows you to view and use the **Twilight threshold exceeding** communication object through which the device notifies that the brightness value detected by the twilight sensor is higher/lower (according to the value settings) than the threshold. Using this object, the device is able to reproduce the typical functions of the twilight sensor which is not combined with the movement sensor.

The settings are:

- **disabled**

The **Twilight threshold exceeding** communication object is not visible and therefore the device will not signal that the twilight sensor has been exceeded.

- **direct (1 if brightness below the threshold)**

When the twilight sensor detects a level of brightness below the set threshold, it sends a bus telegram with a "1" logic value to the communication object; vice versa, when the brightness level detected exceeds the threshold value, the device sends a telegram with a "0" logic value to the same communication object.

- **reverse (0 if brightness below the threshold)**

When the twilight sensor detects a level of brightness below the set threshold, it sends a bus telegram with a "0" logic value to the communication object; vice versa, when the brightness level detected exceeds the threshold value, the device sends a telegram with a "1" logic value to the same communication object.

## 4.2 Communication objects

The communication objects enabled by the **Twilight sensor** menu are those seen in Diag. 4.2.

Number	Name	Object Function	Length	C	R	W	T	U	Data Type	Priority
13	Twilight threshold exceeding	On/Off	1 bit	C	R	-	T	-	1 bit DPT_Switch	Low

Diag. 4.2

### ➤ 4.2.1 *Twilight threshold exceeding*

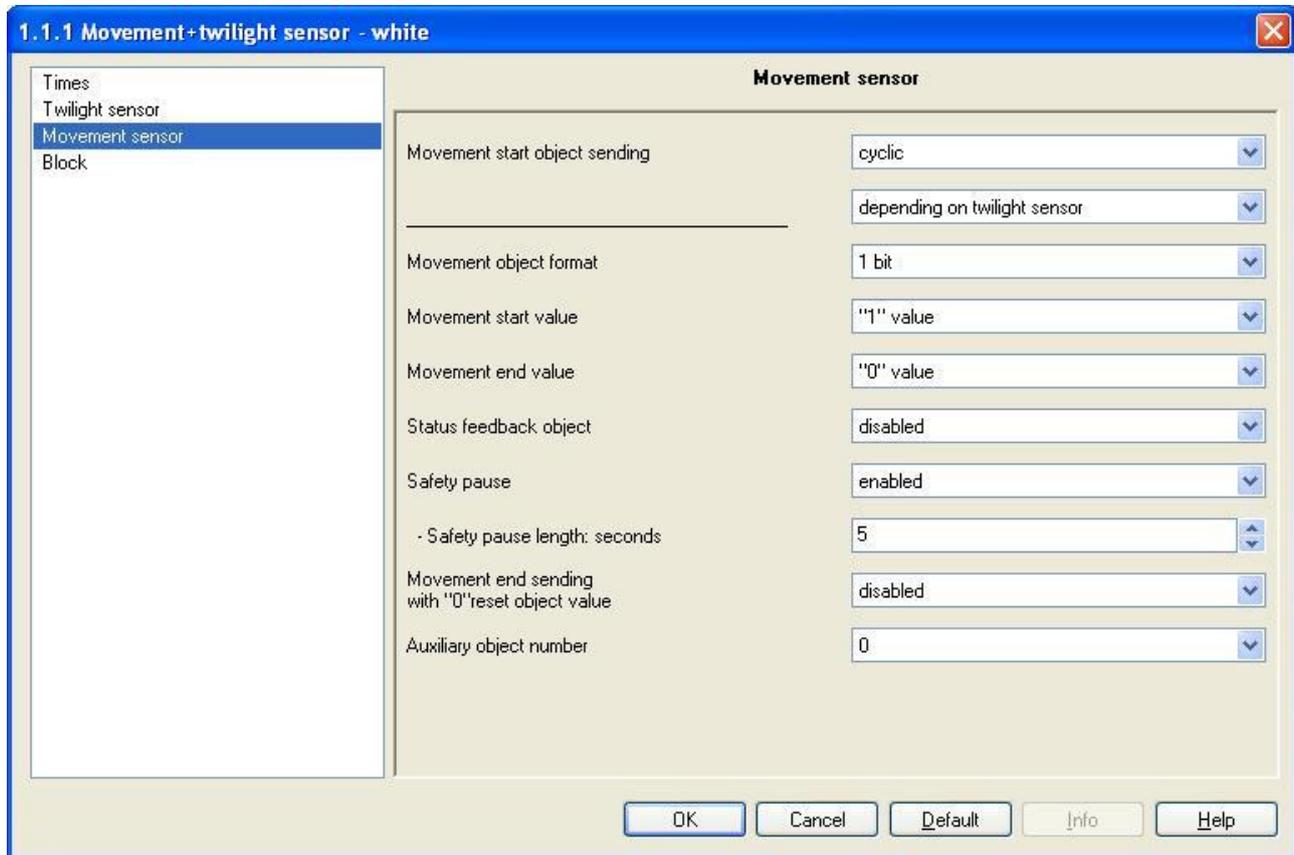
The device sends signals to the bus through this communication object which indicate that the twilight sensor has detected a brightness level that exceeds the twilight sensor; these signals are sent as soon as the detected brightness value exceeds the threshold, or falls below the threshold and the logic values of the telegrams linked to these two events depends on the value set for the **Twilight threshold exceeding object** option.

The enabled flags are C (communication), R (read by bus) and T (transmission) .

The standard format of the object is *1.001 DPT\_Switch*, so the size of the object is *1 bit* and the command it sends is *twilight threshold value exceeding*.

## 5 “Movement sensor” menu

The **Movement sensor** menu lists all the parameters needed to configure the functions of the IR twilight sensor installed on the device; Diag. 5.1 shows the complete **Movement sensor** menu screen shot with the relative configurable parameters.



Diag. 5.1

### 5.1 Parameters

#### ➤ 5.1.1 Movement start object sending

This enables the sending of periodic telegrams linked to the “movement start” event and also subordinate the sending of this telegram according to the brightness value detected by the twilight sensor;

The settings in the first menu window are:

- **cyclic**

the telegram linked to the “movement start” event is sent periodically, at intervals equal to the set cycle time; this means that, each time the cycle time expires, the device will send a "movement start" telegram.

- **not cyclical**

The telegram linked to the “movement start” event is not sent periodically; this means that, each time the cycle time expires, the device will not send a "movement start" telegram as it is only sent when the first movement is detected.

The settings in the second menu window are:

- **depending on twilight sensor**

The device detects the “movement start” event and sends the associated telegram when movement is detected by the IR movement sensor; at the same time the brightness value measured by the twilight sensor is below the threshold, which creates conditioned movement detection.

The condition of brightness measured below the threshold is taken into consideration by the device to assert the "movement start" event only, whilst for all subsequent detection the value measured by the twilight sensor is not taken into consideration as the measurement could easily be influenced by the activation of the loads that the device controls; once that the "movement end" event has been asserted, the value of the twilight sensor again contributes to detecting conditioned movement.

- **not depending on twilight sensor**

The device detects the "movement start" event and sends the associated telegram simply when movement is detected by the IR movement sensor, without taking into consideration the brightness value measured by the twilight sensor.

### ➤ 5.1.2 Movement object format

Here you can configure the format of the bus telegrams which will be sent when the start and movement end events occur.

According to the value set for this item, different value options will be visible for the **Movement start value** and **Movement end value** items. The settings are:

- **1 bit**

The command format that the device will send following a start or movement end event is 1 bit, so it will have a "1" or "0" logic value that, according to how it is used, could for instance perform an ON/OFF command, an UP/DOWN command or a TRUE/FALSE Boolean value command.

- **1 byte**

The command format that the device will send following a start or movement end event is 1 byte and in this case it will be a percentage value.

- **scene**

The command format that the device will send following a start or movement end event is 1 byte and in this case it will be preset execute scene value.

### ➤ 5.1.3 Movement start value

Here you can configure the command or the value to be sent following a movement start event.

According to the value set for the **Movement object format** option, the parameter options also change:

– If the format of the movement object to be sent is **1 bit**, the values to be configured are:

- **no effect**

When the movement start event occurs, the device will not send a telegram to the bus.

- **"0" value**

When a movement start event occurs, the device will send a telegram to the bus with a "0" logic value through the **Movement (switching)** communication object.

- **"1" value**

When a movement start event occurs, the device will send a telegram to the bus with a "1" logic value through the **Movement (switching)** communication object.

– If the format of the object to be sent is **1 byte**, the values to be configured are:

- the values range from 0% to 100% with a 10% pitch plus the **no effect** value. When a movement start event occurs, the device will send a telegram to the bus with the percentage value set for the relative option through the **Movement (% value)** communication object; when the **no effect** option is selected, no telegrams are sent when an event occurs

– If the format of the movement object is **scene**, the values to be configured are:

- **no effect**

When the movement start event occurs, the device will not send a telegram to the bus.

- **scene number**

When a movement start event occurs, the device will send a telegram to the bus with an execute scene command through the **Movement (scene)** communication object. When this value is selected, the new **Scene number** option appears which will be analysed further on.

➤ **5.1.4 Movement end value**

Here you can configure the command or the value to be sent following a movement end event.

For further details please see **5.1.3 Movement start value**, where the difference is that the telegrams are sent when a movement end event occurs.

➤ **5.1.5 Scene number**

Here you can assign the scene number which you intend to execute following a start or movement end event (according to the event the item refers to); the settings range from 0 to 63.

➤ **5.1.6 Status feedback object**

Here it is possible to view and use the **Status feedback** communication object through which the device is able to receive the status signal of the load it controls; this occurs especially when a safety pause is inserted between the moment when the "movement end" occurs and the moment when the device is able to detect new movements. This is because, on enabling this option, the safety pause is calculated from the moment the **Status feedback** communication object receives the signal that the load has been deactivated and not from the movement when the "movement end" is detected.

The settings are:

- **disabled**

The **Status feedback** communication object is not visible and therefore the device is not able to acknowledge the status of the load it controls.

- **enabled**

The **Status feedback** communication object is visible and therefore the device is able to acknowledge the status of the load it controls.

➤ **5.1.7 Safety pause**

This allows you to enable the possibility to enter a safety pause between the moment when the "movement end" is detected and the moment when the device is able to detect new movements; this prevents the deactivation of the load controlled by the device from generating false trigger events on the movement sensor which are mainly caused by the variation in temperature of the illuminating body; in fact during the safety pause, the device is not able to detect new movements. This option is visible if the **Movement end value** option has a value other than **no action** and/or the **Status feedback object** option is set to **enabled**; in the case where the status feedback object is not enabled, the safety pause will commence when the device sends to the bus telegram associated to the "movement end" event, whilst, when the notification option is enabled, remember that the pause begins at the moment that the notification is received that the load has been deactivated.

The settings are:

- **disabled**

The safety pause is not enabled so immediately after a "movement end" event the device is instantly able to detect new movements. With this setting the **Safety pause length : seconds** is not visible.

- **enabled**

The safety pause is enabled so the device is only able to detect new movements when this safety pause time has lapsed. With this setting the **Safety pause length : seconds** option is visible.

➤ **5.1.8 Safety pause length : seconds**

Here it is possible to set the safety pause time in seconds.

The settings range from 1 (second) to 10 (seconds).

➤ **5.1.9 Movement end sending with "0" reset object value**

This enables the possibility to record the "movement end" event when the **Reset** object receives a bus telegram with a "0" logic value.

The settings are:

- **disabled**

When the **Reset** communication object receives a telegram with a “0” logic value, this produces no effect on the device.

- **enabled**

When the **Reset** communication object receives a telegram with a “0” logic value during the cycle time, this triggers a “movement end” event, replicating the same procedures as when the recovery time lapses.

➤ **5.1.10 Auxiliary object number**

Here you can enable new independent auxiliary objects in order to control a series of loads simultaneously; according to the value set for this option, new configuration menus will appear, namely **Aux 1**, **Aux 2** and **Aux 3** that allow you to configure the functions for the relative command objects. The settings range from 0 (auxiliary inputs) to 3 (auxiliary objects).

## 5.2 Communication objects

The communication objects enabled by the **Movement sensor** menu are those seen in Diag. 5.2.

Number	Name	Object Function	Length	C	R	W	T	U	Data Type	Priority
0	Movement (switching)	On/Off	1 bit	C	R	-	T	-	1 bit DPT_Switch	Low
0	Movement (% value)	Value sending	1 Byte	C	R	-	T	-	8 bit unsigned value DPT_Scaling	Low
0	Movement (scene)	Execute scene	1 Byte	C	R	-	T	-		Low
1	Status feedback	On/Off status	1 bit	C	-	W	-	-	1 bit DPT_Switch	Low
3	Reset	Reset movement	1 bit	C	-	W	-	-	1 bit DPT_Switch	Low

Diag. 5.2

➤ **5.2.1 Movement (switching)**

The device sends telegrams to the bus through this communication object which are associated to the “movement start” and “movement end” events if the **Movement object format** is set to a value of **1 bit**, according to the values set for the **Movement start value** and **Movement end value** options.

The enabled flags are C (communication), R (read by bus) and T (transmission).

The standard format of the object is **1.001 DPT\_Switch**, so the size of the object is **1 bit** and the command it sends is a “1”/”0”. *logic value*

➤ **5.2.2 Movement (% value)**

The device sends telegrams to the bus through this communication object which are associated to the “movement start” and “movement end” events if the **Movement object format** is set to a value of **1 byte**, according to the values set for the **Movement start value** and **Movement end value** options.

The enabled flags are C (communication), R (read by bus) and T (transmission).

The standard format of the object is **1.001 DPT\_Scaling**, so the size of the object is **1 byte** and the command it sends is a *percentage value*.

➤ **5.2.3 Movement (scene)**

The device sends telegrams to the bus through this communication object which are associated to the “movement start” and “movement end” events if a **scene** value is set for the **Movement object format**, according to the values set for the **Movement start value** and **Movement end value** options.

The enabled flags are C (communication), R (read by bus) and T (transmission).

The standard format of the object is **18.001 DPT\_SceneControl**, so the size of the object is **1 byte** and the command it sends is an *execute set scene*.

➤ **5.2.4 Status feedback**

Using this communication object, the device receives status feedback bus telegrams from the actuators that manage the loads controlled by the movement detector.

The enabled flags are C (communication) and W (written by bus).

The standard format of the object is **1.001 DPT\_Switch**, so the size of the object is **1 bit** and the command it receives is *load activated/deactivated*.

➤ **5.2.5 Reset**

Using this communication object, the device receives telegrams from the bus that simulate the "movement start" and "movement end" events without the device actually detecting any movement.

When this communication object receives a bus telegram with a "1" logic value, it is interpreted by the device as a movement detection, and therefore it triggers a "movement start" event or it prolongs the cycle time; with regard to the triggering of a "movement start" event, it is important to remember that this depends on the twilight sensor, also in this case the brightness level detected must be below the twilight threshold to trigger the event, similar to when a movement is detected directly by the device.

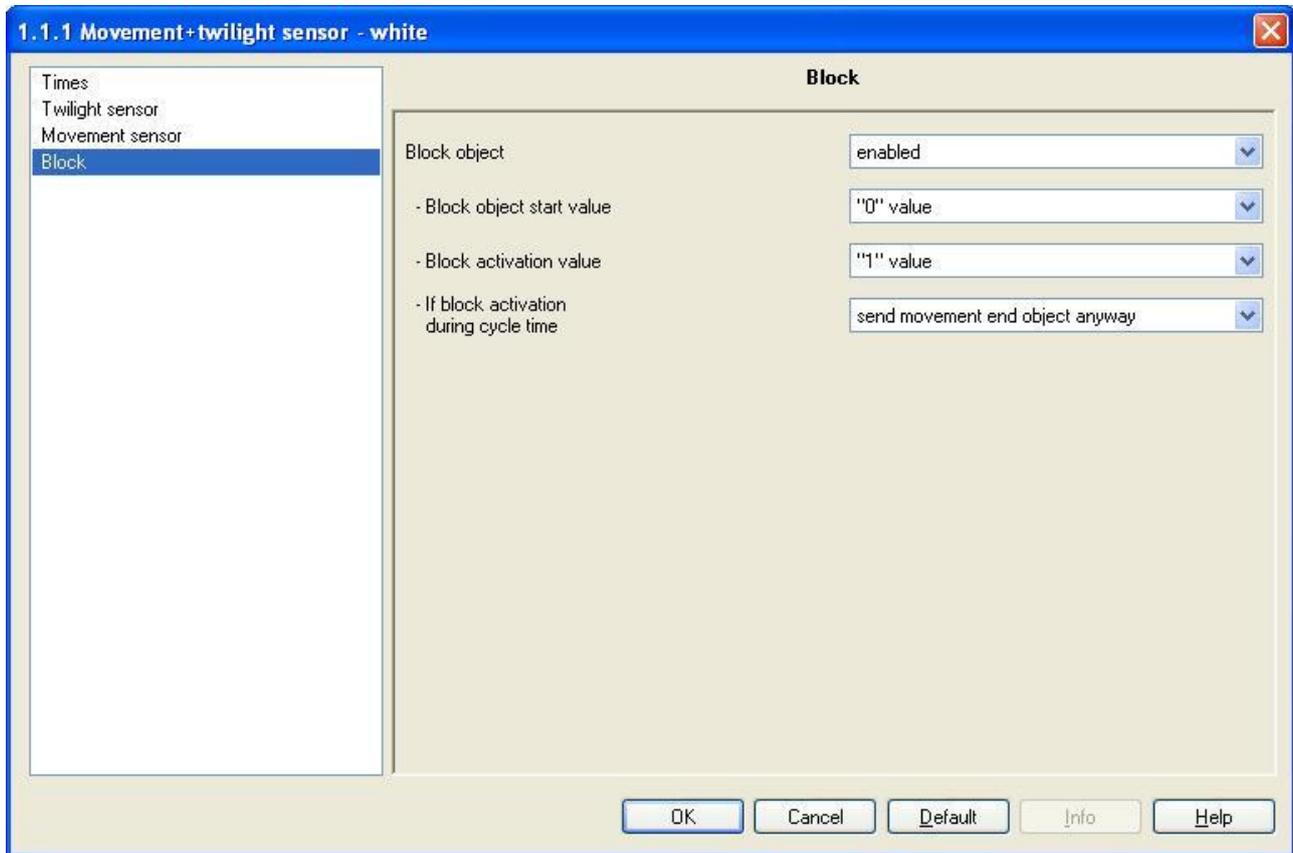
Vice versa, if it is enabled, when this communication object receives a bus telegram with a "0" logic value during the cycle time, it must be interpreted by the device as the recovery time limit, therefore triggering a "movement end" event just as if the recovery time had actually expired.

The enabled flags are C (communication) and W (written by bus) .

The standard format of the object is *1.001 DPT\_Switch*, so the size of the object is *1 bit* and the command it receives is *load activated/deactivated*.

## 6 “Block” menu

The **Block** menu lists all the parameters needed to configure the block functions used by the device, that is the function that prevents the device from operating in normal mode (see Diag. 6.1).



Diag. 6.1

### 6.1 Parameters

#### ➤ 6.1.1 Block object

This is to enable and configure the function and make the (**Block object start value, Block activation value, If block activation during cycle time**) and the relative **Block** communication object visible.

The block function, once the relative trigger command is received from the bus, blocks the device and prevents it from operating in normal mode, that is movement detection. The settings are:

- **disabled**

The block function is not enabled and consequently the various options and communication object are not visible.

- **enabled**

The block function can be enabled by the **Block** communication object and can be activated by a bus command.

In this section you can also view the other settings to configure the function.

#### ➤ 6.1.2 Block object start value

This is used to set what logic value the **Block** communication object must assume each time the bus power is reinstated; the settings are:

- **“0” value**

Each time the bus power is reinstated (29 Volt SELV) the device, on completing the initialisation phase, sets the logic value on the **Block** communication object to “0”; if this is also the block activation value,

once the bus power is reinstated the device is “blocked”, if the block activation value is “1” the device will function normally.

- **“1” value**

Each time the bus power is reinstated (29 Volt SELV) the device, on completing the initialisation phase, sets the logic value on the **Block** communication object to “1”; if this is also the block activation value, once the bus power is reinstated the device is “blocked”, if the block activation value is “0” the device will function normally.

➤ **6.1.3 Block activation value**

This is used to set what logic value the bus telegram must indicate to activate the block function; the settings are:

- **“0” value**

When the device receives a telegram from the bus with a “0” logic value, it activates the block function, preventing the device from operating in normal mode. When the device receives a telegram with a “1”, it deactivates block function if it is activated, on the contrary the command is ignored.

- **“1” value**

When the device receives a telegram from the bus with a “1” logic value, it activates the block function, preventing the device from operating in normal mode. When the device receives a telegram with a “0”, it deactivates block function if it is activated, on the contrary the command is ignored.

➤ **6.1.4 If block activation during cycle time**

This allows you to set the behaviour of the device when it receives a block activation bus telegram during the cycle time; the settings are:

- **send movement end object anyway**

When the device receives a block activation telegram during the cycle time this prevents all further movement detection and also prevents the sending of a “movement end” event bus telegram.

- **does not send movement end object**

When the device receives a block activation telegram during the cycle time this only prevents it from further movement detection.

## 6.2 Communication objects

The communication objects enabled by the **Block** menu are those seen in Diag. 6.2.

Number	Name	Object Function	Length	C	R	W	T	U	Data Type	Priority
2	Block	Switching on/off	1 bit	C	-	W	-	-	1 bit DPT_Enable	Low

Diag. 6.2

➤ **6.2.1 Block**

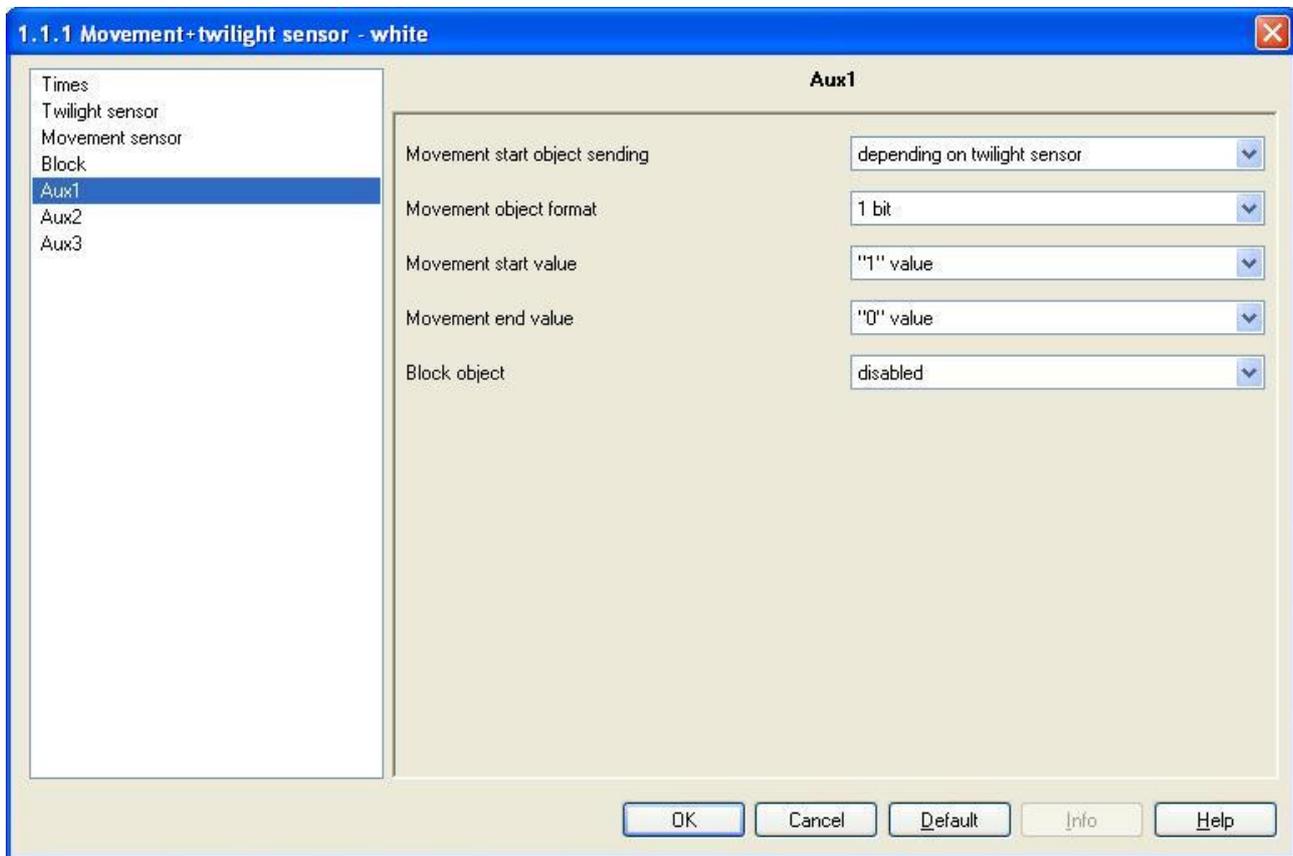
Using this communication object, the device is able to receive the block switching on/off commands from the bus.

The enabled flags are C (communication) and W (written by bus) .

The standard format of the object is *1.003 DPT\_Enable*, so the size of the object is *1 bit* and the information it receives is *block switched on/off* .

## 7 “Aux x” Menu

The **Aux x** menu, that summarises the content of the **Aux 1**, **Aux 2** and **Aux 3** menus, lists all the parameters needed to configure the functions of the IR twilight sensor installed on the device, for what concerns the auxiliary object “X”; Diag. 7.1 shows the complete **Aux x** menu screen shot with the relative configurable parameters.



Diag. 7.1

### 7.1 Parameters

#### ➤ 7.1.1 Movement start object sending

Here you can subordinate the sending of the telegram linked to the “movement start” event to the brightness value detected by the twilight sensor on the general auxiliary object “x”; the settings are:

- **depending on twilight sensor**

The device detects the “movement start” event and sends the associated telegram when movement is detected by the IR movement sensor; at the same time the brightness value measured by the twilight sensor is below the threshold, which creates conditioned movement detection.

The condition of brightness measured below the threshold is taken into consideration by the device to assert the “movement start” event only, whilst for all subsequent detection the value measured by the twilight sensor is not taken into consideration as the measurement could easily be influenced by the activation of the loads that the device controls; once that the “movement end” event has been asserted, the value of the twilight sensor again contributes to detecting conditioned movement.

- **not depending on twilight sensor**

The device detects the “movement start” event and sends the associated telegram simply when movement is detected by the IR movement sensor, without taking into consideration the brightness value measured by the twilight sensor.

### ➤ 7.1.2 Movement object format

Here you can configure the format of the bus telegrams which will be sent when the start and movement end events occur on the general auxiliary object "x".

According to the value set for this item, different value options will be visible for the **Movement start value** and **Movement end value** items. The settings are:

- **1 bit**

The command format that the device will send following a start or movement end event is 1 bit, so it will have a "1" or "0" logic value that, according to how it is used, could for instance perform an ON/OFF command, an UP/DOWN command or a TRUE/FALSE Boolean value command.

- **1 byte**

The command format that the device will send following a start or movement end event is 1 byte and in this case it will be a percentage value.

- **scene**

The command format that the device will send following a start or movement end event is 1 byte and in this case it will be preset execute scene value.

### ➤ 7.1.3 Movement start value

Here you can configure the command or the value, of the general auxiliary object "x" to be sent following a movement start event.

According to the value set for the **Movement object format** option, the parameter options also change:

– If the format of the movement object to be sent is **1 bit**, the values to be configured are:

- **no effect**

When the movement start event occurs, the device will not send a telegram to the bus.

- **"0" value**

When a movement start event occurs, the device will send a telegram to the bus with a "0" logic value through the **AuxX movement (switching)** communication object.

- **"1" value**

When a movement start event occurs, the device will send a telegram to the bus with a "1" logic value through the **AuxX movement (switching)** communication object.

– If the format of the object to be sent is **1 byte**, the values to be configured are:

- the values range from 0% to 100% with a 10% pitch plus the **no effect** value. When a movement start event occurs, the device will send a telegram to the bus with the percentage value set for the relative option through the **AuxX movement (% value)** communication object; when the **no effect** option is selected, no telegrams are sent when an event occurs

– If the format of the movement object is **scene**, the values to be configured are:

- **no effect**

When the movement start event occurs, the device will not send a telegram to the bus.

- **scene number**

When a movement start event occurs, the device will send a telegram to the bus with an execute scene command through the **AuxX movement (scene)** communication object. When this value is selected, the new **Scene number** option appears which will be analysed further on.

### ➤ 7.1.4 Movement end value

Here you can configure the command or the value, of the general auxiliary object "x" to be sent following a movement end event.

For further details please see **7.1.3 Movement start value**, where the difference is that the telegrams are sent when a movement end event occurs.

### ➤ 7.1.5 Scene number

Here you can assign the scene number, of the general auxiliary object “x”, which you intend to perform following a start or movement end event (according to the event the item refers to); the settings range from 0 to 63.

### ➤ 7.1.6 Block object

This is to enable and configure the function and make the general **AuxX\_Block** configuration menu visible.

The block function, once the relative trigger command is received from the bus, blocks the device preventing the general auxiliary object “x” from operating in normal mode, that is movement detection performed by this object. The settings are:

- **disabled**

The block function is not enabled and consequently the **AuxX\_Block** is not visible.

- **enabled**

The block function is enabled and consequently the **AuxX\_Block** is visible and can be configured.

## 7.2 Communication objects

The communication objects enabled by the **Movement sensor** menu are those seen in Diag. 7.2.

Number	Name	Object Function	Length	C	R	W	T	U	Data Type	Priority
7	Aux1 movement (switching)	On/Off	1 bit	C	R	-	T	-	1 bit DPT_Switch	Low
9	Aux2 movement (switching)	On/Off	1 bit	C	R	-	T	-	1 bit DPT_Switch	Low
11	Aux3 movement (switching)	On/Off	1 bit	C	R	-	T	-	1 bit DPT_Switch	Low
8	Aux1 movement (% value)	Value sending	1 Byte	C	R	-	T	-	8 bit unsigned value DPT_Scaling	Low
10	Aux2 movement (% value)	Value sending	1 Byte	C	R	-	T	-	8 bit unsigned value DPT_Scaling	Low
12	Aux3 movement (% value)	Value sending	1 Byte	C	R	-	T	-	8 bit unsigned value DPT_Scaling	Low
8	Aux1 movement (scene)	Execute scene	1 Byte	C	R	-	T	-		Low
10	Aux2 movement (scene)	Execute scene	1 Byte	C	R	-	T	-		Low
12	Aux3 movement (scene)	Execute scene	1 Byte	C	R	-	T	-		Low

Diag. 7.2

### ➤ 7.2.1 AuxX movement (switching)

The devices sends telegrams to the bus through this communication object which are associated to the “movement start” and “movement end” events (related to the general auxiliary object “x”) if the **Movement object format** is set to a value of **1 bit**, according to the values set for the **Movement start value** and **Movement end value** options.

The enabled flags are C (communication), R (read by bus) and T (transmission) .

The standard format of the object is *1.001 DPT\_Switch*, so the size of the object is *1 bit* and the command it sends is a *“1”“0” .logic value*

### ➤ 7.2.2 AuxX movement (% value)

The devices sends telegrams to the bus through this communication object which are associated to the “movement start” and “movement end” events ( related to the general auxiliary object “x”) if the **Movement object format** is set to a value of **1 Byte**, according to the values set for the **Movement start value** and **Movement end value** options.

The enabled flags are C (communication), R (read by bus) and T (transmission) .

The standard format of the object is *1.001 DPT\_Scaling*, so the size of the object is *1 byte* and the command it sends is a *percentage value*.

### ➤ 7.2.3 AuxX movement (scene)

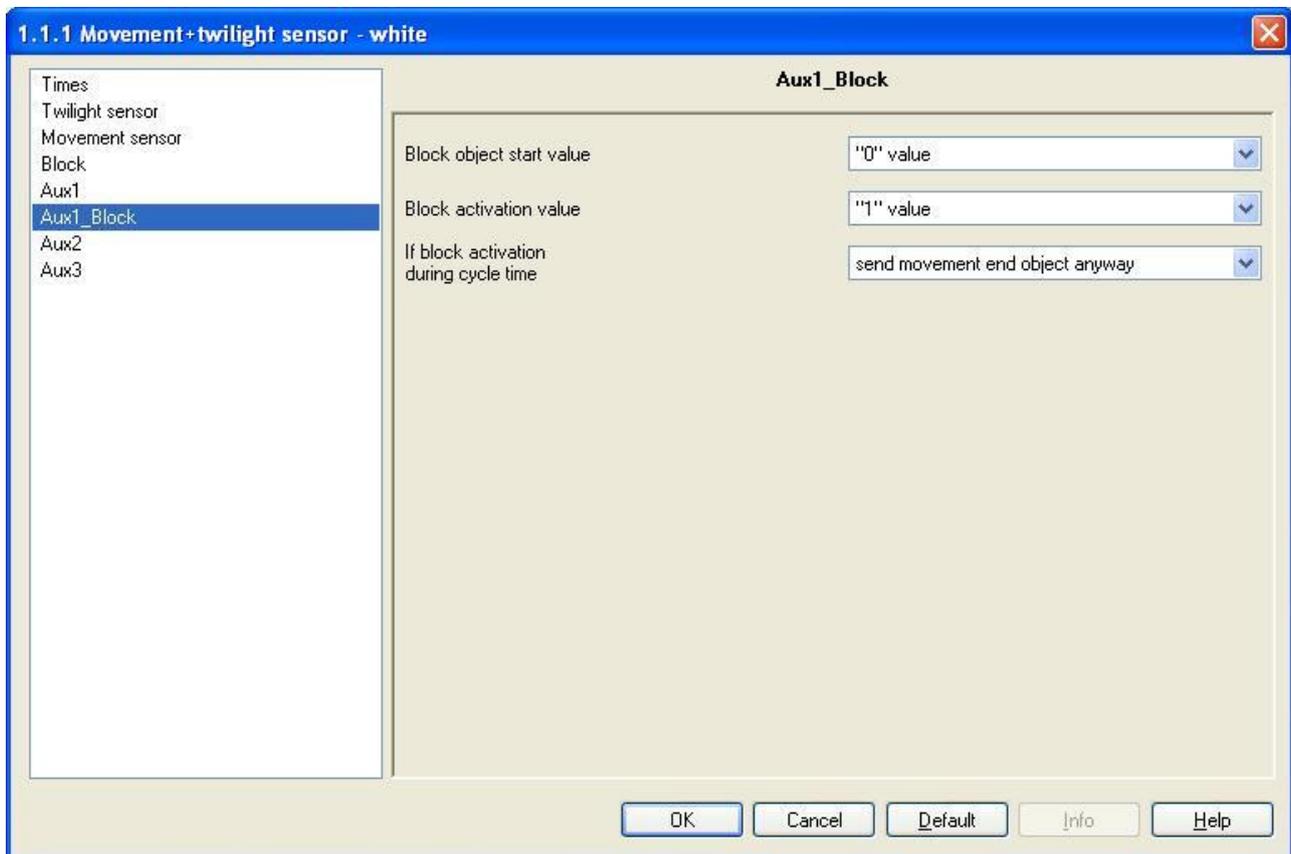
The devices sends telegrams to the bus through this communication object which are associated to the “movement start” and “movement end” events ( related to the general auxiliary object “x”) if the **Movement object format** is set to a **scene** value, according to the values set for the **Movement start value** and **Movement end value** options.

The enabled flags are C (communication), R (read by bus) and T (transmission) .

The standard format of the object is *18.001 DPT\_SceneControl*, so the size of the object is *1 byte* and the command it sends is *execute set scene*.

## 8 “AuxX\_Block” menu

The **AuxX\_Block** menu, that summarises the content of the **Aux1\_Block**, **Aux2\_Block** and **Aux3\_Block** menus, lists all the parameters needed to configure the block function used on the device, that is the function that prevents the general auxiliary object “x”, from operating in normal mode (see Diag. 8.1).



Diag. 8.1

### 8.1 Parameters

#### ➤ 8.1.1 Block object start value

This is used to set what logic value the **AuxX block** communication object must assume each time the bus power is reinstated; the settings are:

- “0” value

Each time the bus power is reinstated (29 Volt SELV) the device, on completing the initialisation phase, sets the logic value on the **AuxX block** communication object to “0”; if this is also the block activation value, each time the bus power is reinstated the general auxiliary channel x will be “blocked”, if the block activation value is “1” the general auxiliary “x” on the device will function normally.

- “1” value

Each time the bus power is reinstated (29 Volt SELV) the device, on completing the initialisation phase, sets the logic value on the **AuxX block** communication object to “1”; if this is also the block activation value, each time the bus power is reinstated the general auxiliary channel x will be “blocked”, if the block activation value is “0” the general auxiliary “x” on the device will function normally.

#### 8.1.2 Block activation value

This is used to set what logic value the bus telegram must indicate to activate the block function for the general auxiliary channel “x”; the settings are:

- **“0” value**

When the device receives a telegram from the bus with a “0” logic value, it activates the block function, preventing the general channel “x” device from operating in normal mode. When the device receives a telegram with a “1”, it deactivates block function if it is activated, on the contrary the command is ignored.

- **“1” value**

When the device receives a telegram from the bus with a “1” logic value, it activates the block function, preventing the general channel “x” device from operating in normal mode. When the device receives a telegram with a “0”, it deactivates block function if it is activated, on the contrary the command is ignored.

➤ **8.1.3 If block activation during cycle time**

This allows you to set the behaviour of the device when it receives an auxiliary channel “x” block activation bus telegram during the cycle time; the settings are:

- **send movement end object anyway**

When the device receives a block activation telegram during the cycle time this prevents all further movement detection and also prevents the sending of a “movement end” event bus telegram which refers to the general channel “x”.

- **does not send movement end object**

When the device receives a block activation telegram during the cycle time this only prevents it from further movement detection on the general channel “x”.

## 8.2 Communication objects

The communication objects enabled by the **AuxX\_Block** menu are those seen in Diag. 8.2.

Number	Name	Object Function	Length	C	R	W	T	U	Data Type	Priority
4	Aux1 block	Switching on/off	1 bit	C	-	W	-	-	1 bit DPT_Enable	Low
5	Aux2 block	Switching on/off	1 bit	C	-	W	-	-	1 bit DPT_Enable	Low
6	Aux3 block	Switching on/off	1 bit	C	-	W	-	-	1 bit DPT_Enable	Low

Diag. 8.2

➤ **8.2.1 AuxX block**

Using this communication object, the device is able to receive the block switching on/off commands, for the general channel “x” to which the object refers, from the bus.

The enabled flags are C (communication) and W (written by bus) .

The standard format of the object is *1.003 DPT\_Enable*, so the size of the object is *1 bit* and the information it receives is *block switched on/off* .

**GEWISS - MATERIALE ELETTRICO**

**SAT**



**+39 035 946 111**  
8.30 - 12.30 / 14.00 - 18.00  
da lunedì a venerdì



**+39 035 946 260**  
24 ore al giorno



**SAT on line**  
[gewiss@gewiss.com](mailto:gewiss@gewiss.com)