



# IRSC Plus

## Air Conditioning Controller

**ZN1CL-IRSC**

Application program version: [6.12], [8.2]

User manual edition: [8.2]\_a

# CONTENTS

---

Contents .....	2
Document updates.....	3
1 Introduction .....	4
1.1 The IRSC Controller .....	4
1.2 Installation.....	5
1.3 The IRSC Plus Application Program .....	6
2 Configuration.....	7
3 ETS Parameterisation .....	8
3.1 Default Configuration.....	8
3.2 General Parameters .....	11
3.3 Status.....	12
3.4 Reset - Initial Configuration.....	14
3.5 Reset - Data Update .....	16
3.6 Scenes.....	17
3.7 Timers.....	19
3.8 Window Sensor .....	20
3.9 Presence Detector.....	21
ANNEX I. Diagnosing the IR Transmitter .....	23
ANNEX II. Communication objects .....	25

## DOCUMENT UPDATES

Version	Changes	Page(s)
[8.2]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> <li>- Adjustment of fan speed ranges.</li> <li>- Version 8.2: Supported from ETS version 4 onwards.</li> </ul>	-
[8.0]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> </ul>	-
[6.12]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> <li>- Switch-on orders while the window is open are now ignored.</li> </ul>	-
[6.11]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> </ul>	-
[6.10]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> </ul>	-
[6.7]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> </ul>	-
[6.4]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> </ul>	-
[6.1]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Additional split models available for control (*).</li> </ul>	-
[5.14]_b	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>- Improved compatibility with models 13 and 18.</li> <li>- Method implemented for diagnosing the IR transmitter.</li> <li>- Additional split models available for control (*).</li> </ul>	-

(\*) Please refer to the updated correspondence table, available at <http://www.zennio.com>

# 1 INTRODUCTION

---

## 1.1 THE IRSC CONTROLLER

---

The **IRSC** controller from Zennio is aimed at controlling air conditioning systems featuring infrared receivers (such as splits, ducted units with an integrated infrared interface, etc) through the emulation of their IR remote control systems. Therefore, IRSC combines the following functions into the same device:

- **Control** of the main functionalities of A/C units (On/Off, temperature, mode, fan speed, etc).
- **Compatibility** with most air conditioning manufacturers (please refer to the [correspondence table](http://www.zennio.com), available at: <http://www.zennio.com>).
- Device **management** through infrared codes:
  - Split units (application program: **IRSC Plus**).
  - A/V systems (application program: **IRSC Open**).
  - Ducted cooling/heating machines, with multiple zones to be acclimated (application program: **IRSC Zone**).



Figure 1. IRSC controller

## 1.2 INSTALLATION

The IRSC controller is connected to the KNX bus via the bus connecting terminals.

Once the controller is provided with power from the KNX bus, both the physical address and the associated application program may be downloaded.

In Figure 2, the connection scheme of IRSC is shown.

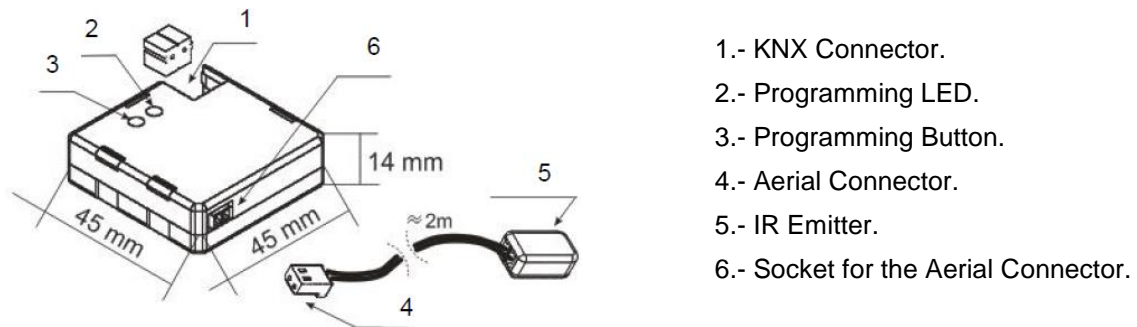


Figure 2. Element Diagram

The functionality of the main elements is described next.

- **Programming Button (3):** a short press on this button sets the device into the programming mode, making the associated LED (2) light in red. If this button is held while plugging the device into the KNX bus, IRSC goes into safe mode.
- **IR emitter (5):** infrared flashing diode to send commands to the air conditioning system. It must be placed in front of the air conditioning IR receiver (see Figure 1.3).
- **Air connector Base (6):** slot for the insertion of the aerial connector, through which the IR commands are outputted from IRSC.

Figure 3 shows how to locate the IR emitter (5) in front of the climate system.

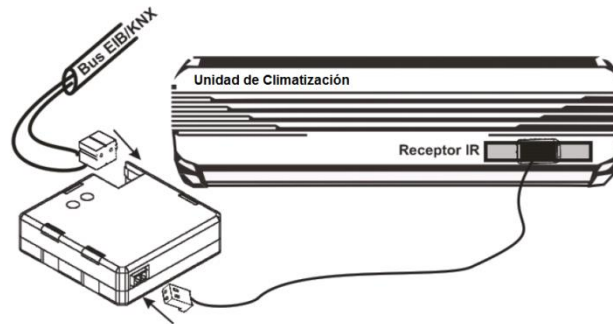


Figure 3. IRSC - A/C Unit Interface

Once IRSC is connected and properly parameterised, it will be possible to control the A/C unit from within the installation (for instance, from the InZennio Z38i or InZennio Z41 touch panels), thus achieving an easier and more intuitive control.

**Note:** *only one air conditioning unit may be controlled by each IRSC.*

To obtain detailed information about the technical features of IRSC, as well as on security and installation procedures, please refer to the device Datasheet, provided with the original packaging of the device and also available at <http://www.zennio.com>.

### 1.3 THE IRSC PLUS APPLICATION PROGRAM

The IRSC Plus application program has been developed to control **common air conditioning systems** via IR commands.

This application allows sending control orders to the A/C system through **infrared commands**, with IRSC acting as a KNX-IR interface, as explained in Section 1.1.

The IR commands required to control the different air conditioning systems have been already preset in the application. It is only necessary to check the IRSC Plus **correspondance table**, available at <http://www.zennio.com>, in order to identify the remote control model associated to the system.

**Important note:** *there are two available versions of IRSC Plus: 6.12 and 8.0. Please, check if the remote control wanted to be emulated with IRSC Plus is available for the version in use. This information is available in the Correspondence Table for IRSC Plus at [www.zennio.com](http://www.zennio.com). In case it were needed to change the current version, just configure and download the right version (please note that a version change might last for several minutes).*

## 2 CONFIGURATION

---

The IRSC Plus application program permits controlling air conditioning systems from different manufacturers as if they were being controlled through the infrared remote control of the machine.

A correspondence table is available at the Zennio homepage (<http://www.zennio.com>) with the codes (from 0 to 255) corresponding to the different remote controls that can be emulated. This number is the first parameter to be set in ETS, so that the orders sent by IRSC become particularised for the actual A/C unit to be controlled.

Moreover, the following functionalities of the A/C system can be configured:

- **ON/OFF** control of the A/C system.
- **Status Objects**, which also work as acknowledgement objects for the orders sent to the system.
- **Working Mode**: the desired working mode (Automatic, Heat, Cool, Fan, Dry) may be chosen through independent binary objects (one per mode) or through a sole object (only one object for the mode selection). Moreover, apart from these conventional controls, it is possible to enable a simplified control, which only permits commuting from Heat to Cold and vice versa.
- **Fan Speed**: step-controlled or through a precise control.
- Possibility of **enabling and configuring**:
  - Up to **4 different scenes**.
  - **Timed ON/OFF** switching.
  - **Energy saving** functions, such as **window sensor** and **presence detection**.

All these options are explained in detail in section 3.

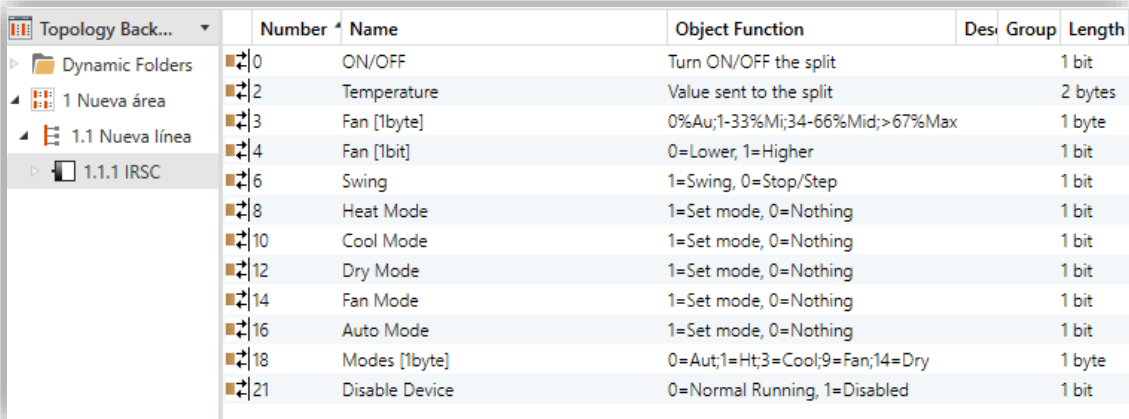
## 3 ETS PARAMETERISATION

To begin with the parameterisation process of the IRSC controller it is necessary, once the data base of the product (**IRSC Plus** application program) has been imported to ETS, to add the device to the project where desired. After that, one right-click on the device will permit selecting "Edit parameters", in order to start the configuration.

In the following sections a detailed explanation can be found about the different functionalities of IRSC Plus in ETS.

### 3.1 DEFAULT CONFIGURATION

This section shows the default configuration the device parameterisation starts from for the first time.



Number	Name	Object Function	Desi	Group	Length
0	ON/OFF	Turn ON/OFF the split			1 bit
2	Temperature	Value sent to the split			2 bytes
3	Fan [1byte]	0%Au;1-33%Ml;34-66%Mid;>67%Max			1 byte
4	Fan [1bit]	0=Lower, 1=Higher			1 bit
6	Swing	1=Swing, 0=Stop/Step			1 bit
8	Heat Mode	1=Set mode, 0=Nothing			1 bit
10	Cool Mode	1=Set mode, 0=Nothing			1 bit
12	Dry Mode	1=Set mode, 0=Nothing			1 bit
14	Fan Mode	1=Set mode, 0=Nothing			1 bit
16	Auto Mode	1=Set mode, 0=Nothing			1 bit
18	Modes [1byte]	0=Aut;1=Ht;3=Cool;9=Fan;14=Dry			1 byte
21	Disable Device	0=Normal Running, 1=Disabled			1 bit

Figure 4. IRSC Plus. Default Topology

The following communication objects are available by default:

- **ON/OFF**, aimed to receive orders from the bus to switch on or off the A/C unit.
- **Temperature**, which will receive setpoint temperatures (in °C) to be sent to the machine. Moreover, this object can work as a status object, as the "R" (read) flag is active by default.



- **Fan [1 byte]**, aimed to receive orders from the bus to switch the A/C machine from one fan speed to another (automatic, minimum, medium or maximum, depending on the range the received value belongs to).

Mode	Received value
Auto	0%
Min	1% - 33%
Middle	34% - 66%
Maximum	> 66%

Table 1. Fan Speed

- **Fan [1 bit]**, aimed to receive orders from the bus to speed up (value “1”) or down (value “0”) the current air flow. This allows stepping through the different fan speed levels sequentially, which will be reflected in the value of the fan status object, if enabled (see section 3.3).
- **Swing**, which is provided for the reception from the bus of orders to control the “swing” function (if available) of the A/C unit. The slats in the output air duct permit leading the air flow towards certain directions, and, depending on the machine, the behaviour may vary slightly. In general, receiving a “1” through this object will make the slats *swing* (i.e., perform a progressive sweep), while one “0” will make them stop. In the case of machines that permit multiple still positions for the slats, the arrival of the value “0” more than once will make IRSC send the A/C unit successive orders to switch from the current position to the next one (*position 1* → ... → *position n*), reverting the sequence order after reaching the last position.
- **Modes [1 byte]**, aimed to receive from the bus orders to switch the A/C unit from one working mode to another one (automatic, heat, cool, fan or dry, depending on the received value).

Mode	Received value
Auto	0 (0x00)
Heat	1 (0x01)
Cool	3 (0x03)
Fan	9 (0x09)
Dry	14 (0x0E)

Table 2. Modes (1 byte)

- A set of independent binary objects, each destined to switching to a certain working mode (**auto**, **heat**, **cool**, **fan** or **dry**) on the reception of the value “1”, as an alternative to the 1-byte Modes object, already described.
- **Disable Device:** binary object that will set IRSC into a standby mode on the reception of the value “1” from the bus. This prevents IRSC from reacting to control orders received from the bus –only read requests will be processed– and from sending IR orders to the A/C unit until the value “0” is received. Therefore, once a certain state has been set to the A/C unit, sending the value “1” to this object will lock that state, preventing bus orders from changing it. This function makes sense, for example, when a centralised control system sends orders simultaneously to several IRSC controllers, and therefore to different A/C units. At a certain time (for power saving reasons, for instance), it may be required that any of them stops reacting to the centralised control orders, which can be done by sending a “1” to its corresponding IRSC controller (through the address of the “Disable device” object).

On the other hand, when IRSC Plus is parameterised for the first time, the following window will be shown:

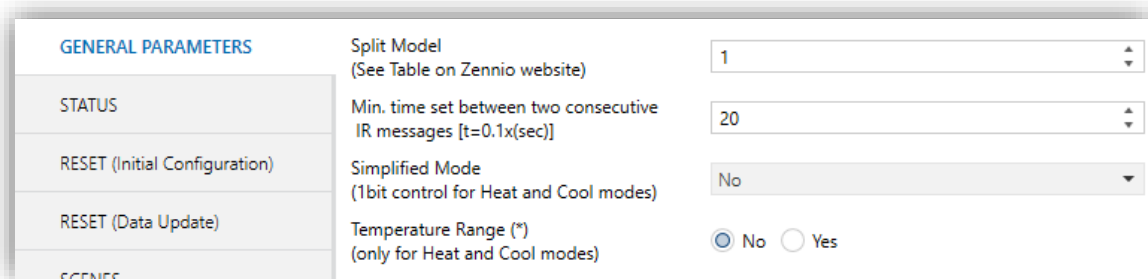


Figure 5. Configuration Screen by default

As shown in Figure 5, the configuration screen is divided into the following main tabs:

- **General Parameters:** general features of the air conditioning system being controlled.
- **Status:** allows enabling and disabling a set of *status* objects over which it will be possible to send read requests from externally. These objects will report information about the current state of the machine and will also work as an

acknowledgement signal of any changes made over this state whenever a control order is received from the bus.

- **Reset:** allows customising the status of the A/C system after a bus power failure or after an application program or parameter download, as well as selecting whether to send or not the updated status objects to the KNX bus.
- **Scenes:** makes it possible to enable and configure up to 4 different climate scenes, or presets.
- **Timers:** allows configuring an automatic switch-off or switch-on of the A/C system after a certain time.
- **Window Sensor:** permits switching the A/C system off when an “open window” signal arrives from the bus, in order to save energy.
- **Presence Detector:** permits switching the A/C system off when a “no presence detection” signal referring to the zone being acclimatised arrives from the bus, in order to save energy.

All the aforementioned tabs are detailed next.

## 3.2 GENERAL PARAMETERS

---

As shown in Figure 5, from the **General Parameters** tab it is possible to configure the main parameters of the A/C system to be controlled. These are the following:

- **Split Model:** sets a code (between 0 and 255) associated to the model of the IR remote of the system to be controlled (please refer to the [correspondence table](http://www.zennio.com) at <http://www.zennio.com>).
- **Min. Time set Between two Consecutive IR Messages:** allows establishing a minimum time gap (in tenths of a second) between consecutive IR commands sent from IRSC Plus to the A/C system (or split), thus restricting the rate of these commands and ensuring the proper reception of all of them. It is recommended to set a value of 20 tenths of a second.

**Note:** *setting a value lower than this does not necessarily imply an improvement in the performance, and may turn into communication issues between the KNX system and the A/C system.*

- **Simplified Mode:** selecting this option brings the 1-bit communication object "Simplified mode", which permits switching from the Cool to the Heat (working modes easily (0=Cool, 1=Heat, or alternatively 0=Heat, 1=Cool). This option is normally intended for simple installations where only binary orders are possible and for climate systems where other modes are not implemented.
- **Temperature Range:** when enabled ("Yes"), this functions allows restricting the values of the setpoint temperatures sent to the machine in the Heat and Cool modes:
  - **Cool mode Min. Temperature:** lower limit, between 16 and 30°C (21°C by default).
  - **Heat mode Max. Temperature:** upper limit, between 16 and 30°C (24°C by default).

Enabling this option will bring a new 1-bit communication object, "Disable temperature limitation", which will need to have the value "1" (default value) in order to have the parameterised temperature limitation effectively applied; in such case, the system will always keep the setpoint over the lower limit for Cool mode and under the upper limit for Heat mode (truncating it when required). While the value of this object is "0", temperature limitation is ignored.

**Note:** *this temperature limitation only applies to modes Heat and Cool, no matter if the simplified mode switching has been enabled or not.*

### 3.3 STATUS

---

The Status tab permits enabling several status communication objects (all of them with the "R" flag enabled by default, making it possible to perform read requests over them). Doing so implicitly makes IRSC Plus confirm the reception of the orders it receives from the KNX bus, so that after transmitting the IR commands to the A/C machine,

IRSC Plus also sends the corresponding status objects to the bus, thus reflecting whether the state has been changed or not as requested.

GENERAL PARAMETERS	Status shall this function be used?	<input type="radio"/> No <input checked="" type="radio"/> Yes
<b>STATUS</b>		
RESET (Initial Configuration)	- ON/OFF	<input checked="" type="radio"/> No <input type="radio"/> Yes
RESET (Data Update)	- Mode	<input checked="" type="radio"/> No <input type="radio"/> Yes
SCENES	- Fan	<input checked="" type="radio"/> No <input type="radio"/> Yes
	- Swing	<input checked="" type="radio"/> No <input type="radio"/> Yes

Figure 6. Status Window

It is possible to activate status objects for the following actions:

- **ON/OFF:** enables the "ON/OFF (status)" binary object, which shows the A/C system operation status: value "1" indicates that the system is active, whereas the value "0" indicates the system is inactive.
- **Mode:** enables one status object per available mode ("Heat Mode (status)", "Cool Mode (status)", "Dry Mode (status)", "Fan Mode (status)" and "Auto Mode (status)") as well as the "Mode [1 byte] (status)" object. The first five objects (1-bit) are set to "1" when their corresponding modes are active, while "0" indicates that the mode is not active. The 1-byte object is set to a certain integer value according to the currently active mode (according to Table 2).
- **Fan:** enables the "Fan (status)" 1-byte object, which shows the current fan speed, in percentage, of the A/C system (according to Table 1).
- **Swing:** enables the "Swing (status)" 1-bit object, which shows the current status of the slats of the A/C system (if available): the value "1" means that the swing is active, while "0" indicates that the slats are still. Note that the swing function may differ slightly depending on the A/C system model. Please refer to section 3.1 for more details.

### 3.4 RESET - INITIAL CONFIGURATION

From the “Reset (initial configuration)” and the “Reset (data update)” tabs, the integrator will be able to set the desired response to be performed by IRSC Plus upon KNX power losses as well as the initial status after a download or a device reset from ETS. On the one hand, the former permits defining the desired status in the air machine after the power recovery and after a download (default and custom values are available for the ON/OFF, Temperature, Mode, Fan and Swing functions), whereas, on its side, the latter permits selecting whether the related communication objects should be sent or not to the KNX bus on power recovery or after a download.

Figure 7. Reset (Initial configuration)

- Under the **default** configuration, after a power recovery IRSC Plus will assume the same status it had (and therefore, the air conditioning machine) before the power loss. This will not happen, however, after a download or a device reset from ETS (the previous status will be lost during the download or the device reset); in such case, IRSC Plus will assume ON/OFF=Off, Temperature=25°C, Mode=Cool, Fan=0%, Swing=Stopped.

**Note:** *although IRSC Plus assumes the status it had before the bus power loss, if during the interruption the A/C machine is externally manipulated or is also affected by the power loss (most models do not automatically switch on after a power loss), inconsistency between IRSC and the A/C unit may occur. To avoid this, it is recommended to enable the “Send reset configuration to the split when reset occurs” parameter (explained below). Note that no orders will be sent to the A/C unit on power recovery unless this parameter is enabled.*

- Under a **Custom** configuration, it is possible to parameterise the following:
  - **ON/OFF**: after a bus failure, a download or a device reset, the A/C system it will be possible to send the A/C system an order to recover the last saved status<sup>1</sup> before the power loss or be switched off or on.
  - **Temperature**: after a bus failure, a download or a device reset, the it will be possible to send the A/C system an order to recover the last setpoint temperature (last saved position<sup>1</sup>) or a preset value ("Initial temperature", in Celsius degrees).
  - **Mode**: after a bus failure, a download or a device reset, the A/C system may be set to the last registered mode (last saved status<sup>1</sup>) or to one of the following: Heat, Cool, Auto, Dry or Fan.
  - **Fan**: after a bus failure, a download or a device reset, the A/C system may be required to recover the last registered fan level (last saved status<sup>1</sup>) or a particular one: Auto, Low, Medium or High.
  - **Swing**: after a bus failure, a download or a device reset, the status of the slats can be set to: Last saved status<sup>1</sup>, Swing or Stopped.

Note that for both, a default and for a custom configuration, the “**Send Reset Configuration to the Split (when reset occurs)**” parameter needs to be enabled to actually send the corresponding initial values to the A/C system on the power recovery. If disabled, such IR commands will not be sent, but IRSC will still assume the parameterised configuration as the initial state after the power recovery.

**Important:** *it is recommended, in any case, to enable this sending to avoid inconsistency between the state assumed by IRSC and that actually acquired by the A/C unit once the power failure is solved. On the other hand, sending the corresponding communication objects to the bus (so that the other devices in the KNX environment are properly kept up to date) may be enabled from the “Reset (Data Update)” screen (see Section 3.5)*

---

<sup>1</sup> This specific option has no effect after downloads or device resets because the previous status of the device is always lost; default values (ON/OFF=Off, Temperature=25°C, Mode=Cool, Fan=0%, Swing=Stopped) will be then assumed

### 3.5 RESET - DATA UPDATE

Independently of the status of IRSC Plus after bus power interruptions, a download or a device reset from ETS, and also independently of whether the related orders should be sent or not to the air machine (see Section 3.4), it is possible to activate initial sendings of the status objects to the KNX bus, so that other KNX devices are informed about the (new or not) state of the IRSC Plus controller (and thus, of the state sent to –or assumed in– the A/C system) after a bus interruption, a download or a device reset. This is done from the “Reset (Data Update)” screen.

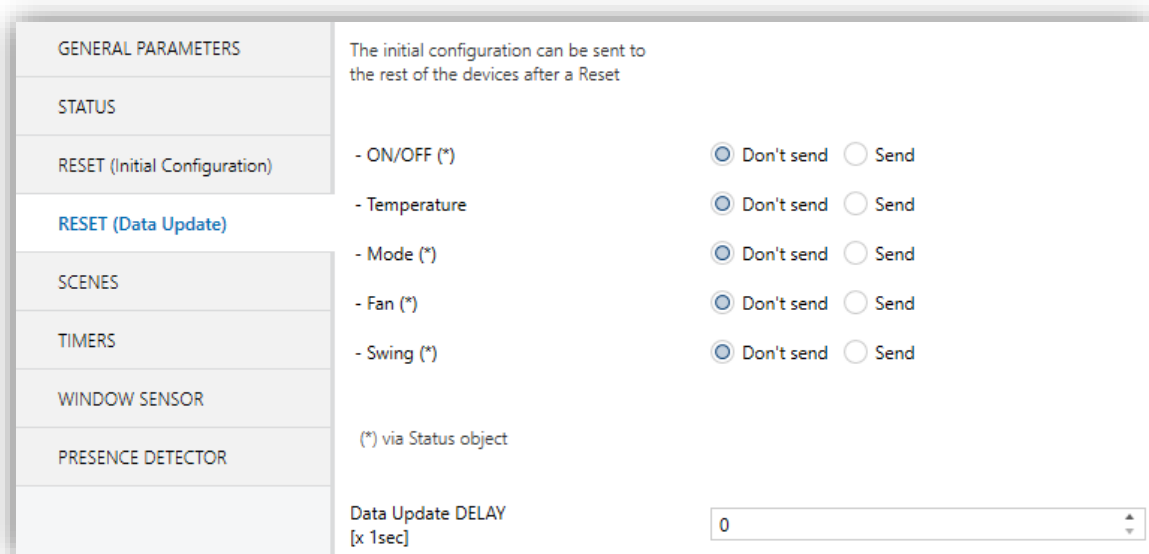


Figure 8. Reset (Data Update)

It is possible to select which status objects will be sent to the bus (“Send”) and which will not (“Don’t send”), among the following:

- **ON/OFF:** the On/Off status of the A/C system will be sent to the bus through the "ON/OFF (status)" object.
- **Temperature:** the setpoint assumed after the bus failure, the download or the device reset will be sent to the bus through the "Temperature" object.
- **Mode:** the working mode assumed by the A/C system after the bus failure, the download or the device reset will be sent to the bus through the "Mode (status)" object.



- **Fan:** the fan level assumed by the A/C system after the bus failure, the download or the device reset will be sent to the bus through the "Fan (status)" object.
- **Swing:** the slats status assumed by the A/C system after the bus failure, the download or the device reset will be sent to the bus through the "Swing (status)" object.

As stated, these values are sent to the KNX bus through the corresponding status object (the status of the temperature setpoint, however, is sent through the "Temperature" general object, also intended for the reception of the control orders). Therefore, **the required status objects should have been enabled (see section 3.3) in order to have these values effectively sent to the bus.**

**Important:** *if this sending is not enabled, the KNX bus will definitively not be informed about the initial status of IRSC Plus, no matter if it is a custom status or the status it already had before the bus failure (see Section 3.4).*

To sum up, selecting "Send" for the above parameters (some or all of them) will make IRSC send its initial status to the rest of the devices connected to the bus, to have them properly updated.

## 3.6 SCENES

---

Scenes consist in simultaneously sending various IR commands to the air conditioning system, so that a certain climate configuration can be activated by simply sending the proper value (i.e., the numeric value associated to that particular scene) through the KNX bus.

IRSC Plus allows enabling and configuring up to **4 different scenes**.

GENERAL PARAMETERS	Scenes (Will scenes be used?)	<input type="radio"/> No <input checked="" type="radio"/> Yes
STATUS		
RESET (Initial Configuration)	Scene 1	<input type="radio"/> No <input checked="" type="radio"/> Yes
RESET (Data Update)	- Scene Number	1
SCENES	- ON/OFF	No Change
TIMERS	- Temperature	No Change
WINDOW SENSOR	- Mode	No Change
PRESENCE DETECTOR	- Fan	No Change
	- Swing	No Change
	Scene 2	<input checked="" type="radio"/> No <input type="radio"/> Yes
	Scene 3	<input checked="" type="radio"/> No <input type="radio"/> Yes
	Scene 4	<input checked="" type="radio"/> No <input type="radio"/> Yes

Figure 9. Scenes Window

After selecting the option "Yes" for "Scenes (will scenes be used?)", the "Scenes [1 byte]" object will be shown. This is the object intended to receive from the KNX bus the scene activation orders.

Four additional fields will appear, from which it is possible to independently enable ("Yes") the four available scenes. For every enabled scene, the following options should be parameterised according to the atmosphere to be generated.

- **Scene Number:** the numeric value (1 to 64) on whose arrival from the bus (through the "Scenes [1 byte]" object, decreased by one –0 to 63–) the air conditioning system will execute the scene.
- **ON/OFF:** permits selecting whether, on the execution of the scene, the A/C system should be switched on ("ON"), off ("OFF") or left as is ("No change").
- **Temperature:** permits selecting whether, on the execution of the scene, the setpoint temperature of the A/C system should be changed to a certain value (between 16 and 30°C) or left as is ("No change").
- **Mode:** permits selecting whether, on the execution of the scene, the A/C system should be set to a certain mode (Auto, Heat, Dry, Fan or Cool) or left as is ("No change").

- **Fan:** permits selecting whether, on the execution of the scene, the fan level of the A/C system should be set to a certain value (Auto, Low, Medium, High) or left as is (“No change”).
- **Swing:** permits selecting whether, on the execution of the scene, the swing function of the A/C system should be set to a certain status (“Stop” or “Swing”) or left as is (no change).

### 3.7 TIMERS

This function permits timed on/off switches of the air conditioning system.

Figure 10. Timers Screen

Automatic switching on and off can be independently enabled and configured. After activating the Automatic OFF, the "Fixed Auto-OFF [1 bit]" object will appear, and the following parameters will also be shown in ETS:

- **Automatic OFF Value:** sets the time interval (between 0 and 255) that IRSC Plus waits once the “Fixed Auto-OFF” object is activated (value "1) before proceeding with the switch-off.
- **Automatic OFF Time Units:** sets whether the above value is expressed in hours or minutes.

After activating the Automatic ON, the "Fixed Auto-ON [1 bit]" object will appear, and the following parameters will also be shown in ETS:

- **Automatic ON Value:** sets the time interval (between 0 and 255) that IRSC Plus waits once the "Fixed Auto-ON" object is activated (value "1") before proceeding with the switch-on.
- **Automatic ON Time Units:** sets whether the above value is expressed in hours or minutes.

### 3.8 WINDOW SENSOR

After activating this feature the 1-bit object "Window sensor" is shown, with the aim to be linked (via the same group address) to any binary input in the installation that monitors whether a window has been opened in the air conditioned zone.

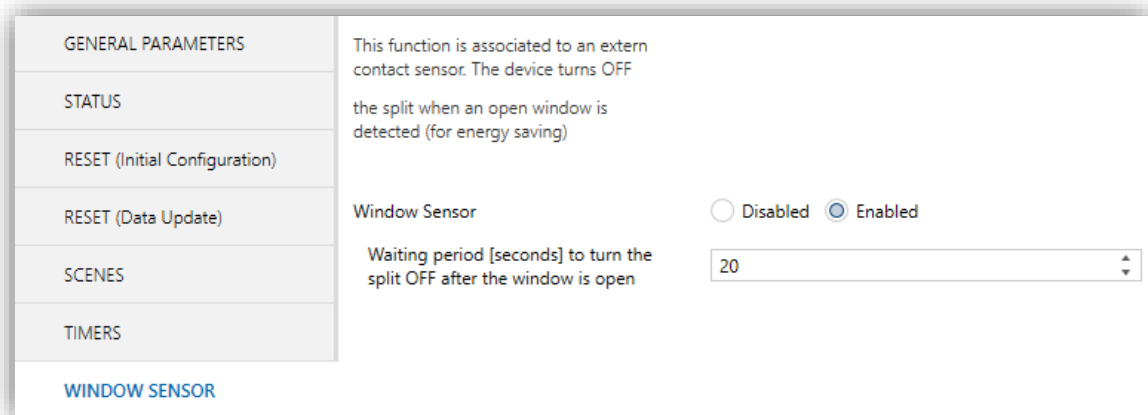


Figure 11. Window Sensor

After activating this function, the "**Waiting Period**" parameter can be configured. It shows the time interval (in seconds) that the window must remain open before the A/C system switches off.

This way, once IRSC Plus receives through the "Window sensor" object the value "1" (value associated to the "open window" event) it will wait for the configured waiting period and then it will send a switch-off order to the A/C system. When the window is closed again ("Window sensor"=0), the A/C system will be automatically switched on – it will not be necessary to send a switch-on order.

**Note:** during the "open window" state, the device will still accept requests from the bus and update its objects according to them. Although the machine is off, as soon as it is switched on again, IRSC will send it the proper commands to update its state as

corresponding. Because of this, in the particular case of receiving a switch-off order while the window is open (therefore, being the machine already off), once the window is closed again, IRSC will not make the A/C unit turn on again – in such case, an explicit ON order will be required. On the other hand, if IRSC receives an order to switch it on again while the window is open (therefore, being the machine already off), it will ignore the order and respond with the “off” value through the status object.

### 3.9 PRESENCE DETECTOR

Presence detectors installed inside a room, permit multiple applications, such as switching lights on and off, detecting intruders, dimming lights, etc. Besides, if combined with the IRSC controller, presence detectors contribute to energy saving by switching the A/C system off when the room is empty.

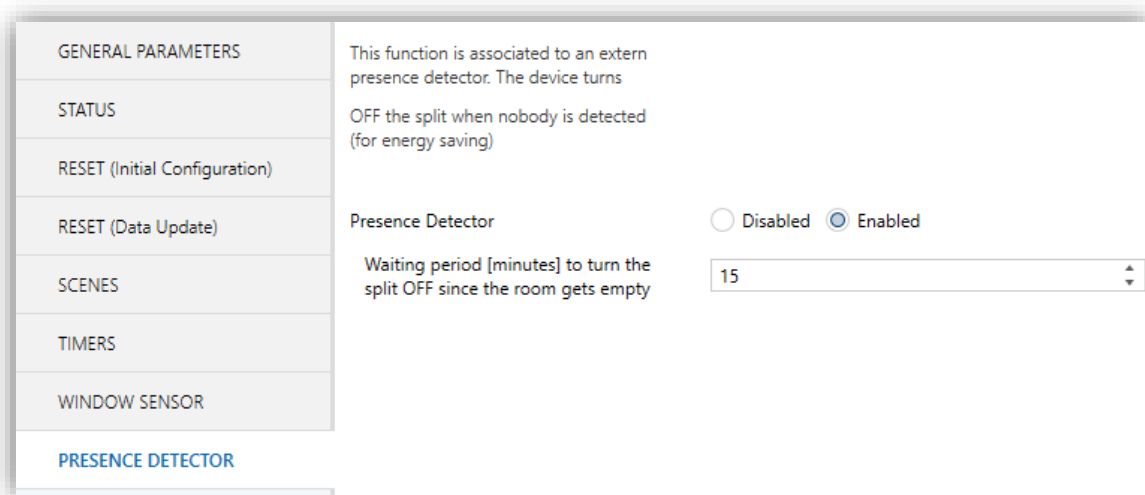


Figure 12. Presence Detector

Activating this feature brings the 1-bit "Presence detection" object, which should be linked to the presence detection object, so that it is aware of presence/no presence changes.

Also the "**Waiting Period**" parameter becomes visible, and defines the time period (in minutes) that IRSC Plus should wait after receiving the value "0" (empty room) through the detection object before switching the A/C unit off.

As soon as the "Presence detection" object receives the value "1", IRSC Plus considers the room is again occupied, so if the value "0" arrives again, the time count will start again.

**Note:** *during the "no detection" state, the device will still accept requests from the bus and update its objects according to them. Although the machine is off, as soon as it is switched on again, IRSC will send the proper commands to update its state as corresponding. Because of this, in the particular case of receiving a switch-off order under "no detection" (therefore, being the machine already off), a later presence detection will not make the A/C unit turn on again as IRSC will then send an order to switch it off. In such case, an explicit ON order will be required.*

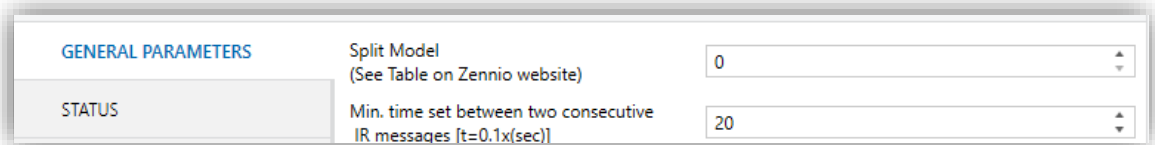
## ANNEX I. DIAGNOSING THE IR TRANSMITTER

This annex describes a method for testing that the IRSC Plus built-in infrared transmitter is working and therefore is properly emulating the remote control of the air conditioning unit and, thus, sending the corresponding orders to the unit.

The procedure explained below consists in emulating, from IRSC Plus, presses made on the remote control (models ZN11R-ZAS and ZN11RZ38) of any of the various Zennio devices capable of receiving IR signals (ZAS, InZennio Z38i, ACTinBOX Hybrid, ACTinBOX MAX6 or ACTinBOX QUATRO). One of these devices will be therefore required during the test.

### Suggested procedure

- Set the value “0” to the **Split model** parameter in the “General parameters” screen.



- Link the ON/OFF object (at least) to a group address.

OBJECT	NAME	DESCRIPTION	GROUP	LENGTH
0	ON/OFF	Turn ON/OFF the split	1/1/3	1 bit
2	Temperature	Value sent to the split		2 bytes

- After downloading parameters and group addresses, IRSC Plus will respond to the **ON/OFF orders** by emitting infrared signals similar to those emitted by the ZN11R-ZAS and ZN11RZ38 remotes when the upper left pair of buttons are pressed (left button after an OFF order; right button after an ON order).

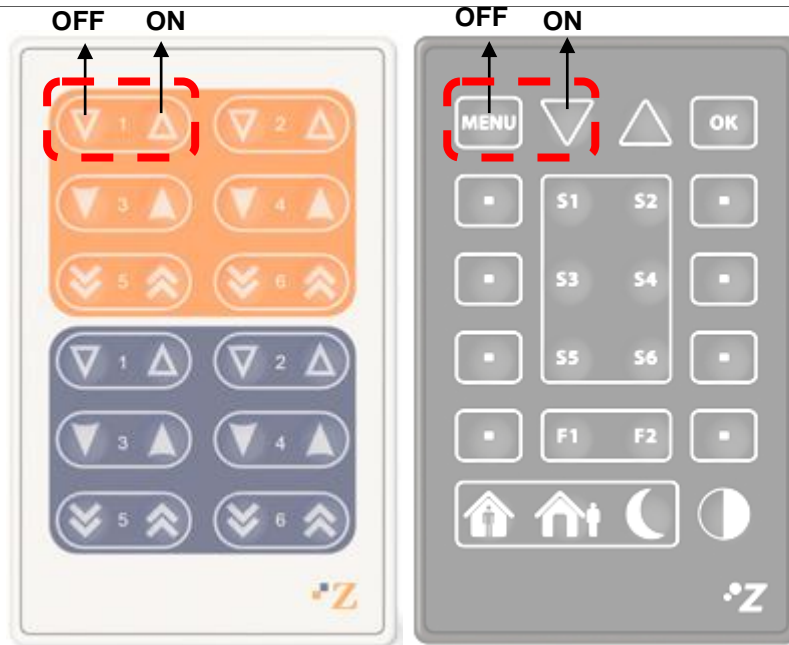


Figure 13 Switch-on and Switch-off orders

This way, sending switch-on and switch-off orders through the ON/OFF object makes it possible to check the effect, over the receiver device (ZAS, InZennio Z38i, ACTinBOX Hybrid, ACTinBOX MAX6 or ACTinBOX QUATRO), of the signals generated by the emitter, and therefore whether IRSC Plus is behaving anomalously or not.

**Important:** *it is highly advisable to locate the IRSC Plus infrared transmitter in front of the optimal reception point in the receiver. In case of using an ACTinBOX actuator as a receiver, the transmitter should be directly attached to the device housing.*



## ANNEX II. COMMUNICATION OBJECTS

- “**Functional range**” shows the values that, with independence of any other values permitted by the bus according to the object size, may be of any use or have a particular meaning because of the specifications or restrictions from both the KNX standard or the application program itself.

Number	Size	I/O	Flags	Data Type (DPT)	Functional Range	Name	Function
0	1 Bit	I	<b>C--WU</b>	DPT_Switch	0/1	ON/OFF	Turn ON/OFF the split
1	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	ON/OFF (status)	Split state (ON/OFF)
2	2 Bytes	I/O	<b>CTRWU</b>	DPT_Value_Temp	-273.00 – 670760.00	Temperature	Value sent to the split
3	1 Byte	I	<b>CT-WU</b>	DPT_Scaling	0% - 100%	Fan [1byte]	0%Au;1-33%Mi; 34-66%Mid;>67%Max
4	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	Fan [1bit]	0=Lower, 1=Higher
5	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	Fan (status)	Actual Fan Level
6	1 Bit	I	<b>CT-WU</b>	DPST-1-1 DPST-1-10	0/1	Swing	1=Swing, 0=Stop/Step
7	1 Bit	O	<b>CTR--</b>	DPST-1-1 DPST-1-10	0/1	Swing (status)	Swing state: 0=Stopped,1=Swing
8	1 Bit	I	<b>CT-WU</b>	DPT_Switch	0/1	Heat Mode	1=Set mode, 0=Nothing
9	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	Heat Mode (status)	1=Heat mode is Set, 0=Not Set
10	1 Bit	I	<b>CT-WU</b>	DPT_Switch	0/1	Cool Mode	1=Set mode, 0=Nothing
11	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	Cool Mode (status)	1=Cool mode is Set, 0=Not Set
12	1 Bit	I	<b>CT-WU</b>	DPT_Switch	0/1	Dry Mode	1=Set mode, 0=Nothing
13	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	Dry Mode (status)	1=Dry mode is Set, 0=Not Set
14	1 Bit	I	<b>CT-WU</b>	DPT_Switch	0/1	Fan Mode	1=Set mode, 0=Nothing
15	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	Fan Mode (status)	1=Fan mode is Set, 0=Not Set
16	1 Bit	I	<b>CT-WU</b>	DPT_Switch	0/1	Auto Mode	1=Set mode, 0=Nothing
17	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	Auto Mode (status)	1=Auto mode is Set, 0=Not Set
18	1 Byte	I	<b>CT-WU</b>	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	Modes [1byte]	0=Aut;1=Ht;3=Cool;9=Fan;14=Dry
19	1 Byte	O	<b>CTR--</b>	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	Modes [1byte] (status)	Actual Mode:0=Auto,1=Heat,etc.

20	1 Byte	I	<b>C - - W U</b>	DPT_SceneControl	0-63; 128-191	Scenes [1byte]	0-63(Run(1-64));128-191(Learn)
21	1 Bit	I/O	<b>C - R W U</b>	DPT_Enable	0/1	Disable Device	0=Normal Running, 1=Disabled
22	1 Bit	I/O	<b>C - R W U</b>	DPT_Start	0/1	Fixed Auto-OFF [1bit]	1=Sw.OFF after "param. value"
23	1 Bit	I/O	<b>C - R W U</b>	DPT_Start	0/1	Fixed Auto-ON [1bit]	1=Sw. ON after "param. value"
24	1 Bit	I	<b>C - - W U</b>	DPT_Bool	0/1	Presence Detection	0=Empty Room, 1=Presence
25	1 Bit	I	<b>C - - W U</b>	DPT_Window_Door	0/1	Window Sensor	1=Window Open, 0=Closed
26	1 Bit	I	<b>C - - W U</b>	DPT_Heat_Cool	0/1	Simplified Mode	0=Heat, 1=Cool
	1 Bit	I	<b>C - - W U</b>	DPT_Heat_Cool	0/1	Simplified Mode	0=Cool, 1=Heat
27	1 Bit	I/O	<b>C - R W U</b>	DPT_Enable	0/1	Disable Temperature Limitation	0=Disabled, 1=Enabled

Join and send us your inquiries  
about Zennio devices:

<http://suppot.zennio.com>

**Zennio Avance y Tecnología S.L.**

C/ Río Jarama, 132. Nave P-8.11  
45007 Toledo (Spain).

*Tel. +34 925 232 002.*

*Fax. +34 925 337 310.*

*www.zennio.com*

*info@zennio.com*



RoHS