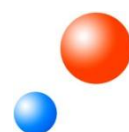




KES

KNX Energy Saver

ZN1IO-KES



Programme version: 1.0

Manual edition: a

INDEX

1. Introduction	3
1.1. KES Energy Saver	3
1.2. Installation	4
1.2.1. Measuring probe	5
2. Configuration	6
2.1. Active Power	6
2.2. Active Energy	7
2.3. Power Limits	7
3. ETS Parameterization	9
3.1. Default configuration	9
3.2. General window	10
3.3. Channels	14
3.3.1. Active Energy	15
3.3.2. Active Power	18
3.3.3. Power Limit Monitoring	20
3.4. Logical functions	23
Annex I. Time management	24
Annex II. Further Examples	27
Annex III. Communication Objects	40

1. INTRODUCTION

1.1. KES ENERGY SAVER

Zennio's 3-channel energy saver, **KES**, combines in the same device the following features:

- 🌐 **3 independent input channels.** Each of them can be used to connect a measuring probe (model **ZN1AC-CST60**) and to perform these actions:
 - Measuring and reporting instant active power consumptions (kW).
 - Measuring and reporting active energy consumptions (kWh).
 - Calculating and reporting the cost of the energetic consumption (with the possibility of setting up **1 to 4 different tariffs**) as well as the volume of the CO₂ emissions required to generate such energy.
 - Monitoring power limits.
- 🌐 Hourly, daily, weekly and monthly report of total energy consumption, as well as upon request (through a **global request** object).
- 🌐 Sending of the **actual and average values** of the consumed energy, its cost and the corresponding CO₂ emissions (every day, every week, every month or upon request).
- 🌐 Support for **single-phase** and **three-phase** environments (two independent application programmes).
- 🌐 Synchronisation capabilities through an external **KNX clock**.
- 🌐 Module with up to 5 multi-operation **logical functions**.



Figure 1.1. KES Energy Saver

1.2. INSTALLATION

KES Energy Saver connects to the KNX bus via the bus connecting terminals.

The input channels are connected to KES through the terminal block with screws included in the device packaging.

Once KES is provided with power supply from the KNX bus, both the individual address and the associated application programme can be downloaded.

The connection scheme for KES is shown on Figure 1.2:

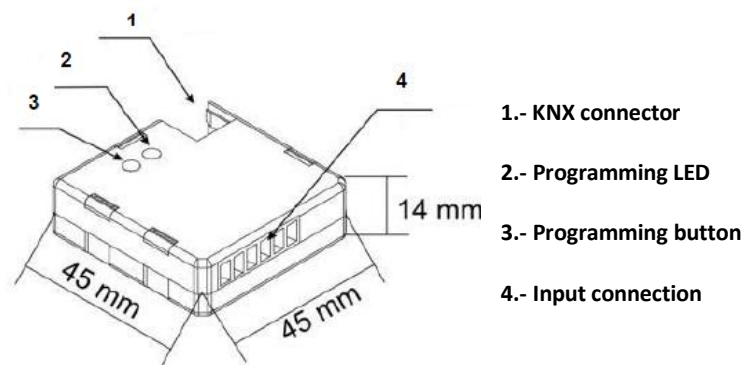


Figure 1.2. KES. Connection scheme

The functionality of the main elements is described below.

- 🎯 **Programming button:** a short press on this button sets the device to programming mode, making the associated LED (2) light in red. If this button is held while plugging the device into the KNX bus, KES goes into secure mode. The LED blinks red.
- 🎯 **Input connection (4):** the terminal block with screws must be plugged here (see Figure 1.3) to allow connecting the different input channels (A, B, C) to the device. One measuring probe will be needed for each channel to be connected. Both terminals of each probe must be inserted in the corresponding input points of the terminal block, depending on the channel desired. For example, when intending to use the three input channels (A, B, C), one probe will have to be connected to points 1 and 2 (the ones related to channel A), another probe to points 3 and 4 (related to channel B) and a third probe to points 5 and 6 (related to channel C).

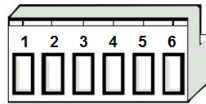


Figure 1.3. Input terminal block

Note: *The insertion order of the two terminal wires from the measuring probe to the connection points of the KES channel does not affect the functionality, i.e., for channel A it is possible to connect either the black wire from the probe into point 1 and the white cable into point 2, or either in the inverse order.*

1.2.1. MEASURING PROBE

The set-up process of the **ZN1AC-CST60** measuring probe (Figure 1.4) is described next.



Figure 1.4. Zennio ZN1AC-CST60 measuring probe

These are the required steps:

- Open the **top clamp** of the probe.
- The **phase wire** from which the energy measures will be taken must be now passed through the gap that has just been uncovered in the probe.
- **Close** back the top clamp of the probe.
- Connect the **two wires** of the measuring probe into the terminal block from KES, in the two connection points corresponding to the channel being used.

Note I: It is **very important** to keep in mind that no ordinary power supplies (230 V) or other external supplies should be connected to either point of the KNX bus, nor directly to the terminal block from KES.

Note II: KES Energy Saver does only measure the energy consumed by **sinusoidal alternating current (AC)** lines.

In order to obtain more detailed information about KES technical features, as well as on security and installation procedures, please read the KES **Datasheet**, included with the original packaging of the device and also available at: <http://www.zennio.com>.

2. CONFIGURATION



2.1. ACTIVE POWER

KES Energy Saver calculates the value of the active power consumed by the device whose phase line passes through the ZN1AC-CST60 probe, connected to one of the KES channels. This active power is obtained from the **effective intensity** measured by the probe according to the following relation:

$$P_{act} = V_{ef} \cdot I_{ef} \cdot \cos \varphi$$

where V_{ef} is the effective voltage of the line and $\cos \varphi$ is the power factor. Both values must be defined by parameter. The value of the effective intensity, I_{ef} , is obtained, as indicated above, once the measuring probe is located around the electric line, and after the outputs of the probe are plugged into one of KES input channels.

The measuring probe, and consequently KES, is able to measure effective intensities in the range [0-60 A].

In addition to measuring instant active power, KES can also monitor the highest detected power, so that **power peaks** can be calculated per hour, day, week and month, which may be interesting for history managing purposes.

2.2. ACTIVE ENERGY

Based on successive values of the instant power, KES Energy Saver can manage the count of the **total energy consumed** per hour, day, week or month, as well as calculating daily, weekly and monthly average consumptions.

KES can also make **estimations of the energy** that would be consumed during a whole day if the current value of the instant power were kept constant for 24 hours. This estimation can be expressed in terms of consumption (kWh) and in terms of cost (local currency).

Daily, weekly and monthly values of the actual energy consumed can be expressed in terms of consumption (kWh), of cost (local currency), or even in terms of the CO₂ (kgCO₂) sent to the atmosphere in order to produce such quantity of energy.

If both, estimations and energy consumption, are set to be expressed in terms of cost, KES allows specifying up to four different tariff values (in cents of the local currency per kWh).

2.3. POWER LIMITS

KES can also monitor the level of the consumed power in relation to certain limits, initially defined by parameter: **upper limit** and **lower limit** (linked to an overconsumption alarm and to a low-consumption indicator, respectively), and a certain margin (or **deadband**) respect to each of the two limits.

Figure 2.1 shows an example of the process. When the instant power is higher than the upper limit, KES sends an overconsumption alarm (if configured) and this will not be deactivated until the instant power is back again lower than the upper limit minus the margin band. If the value of the instant power keeps decreasing, KES will send (if configured) a low-consumption indicator as soon as the power is lower than the lower limit. This indicator will remain active until the power is back again higher than the lower limit plus the margin band.

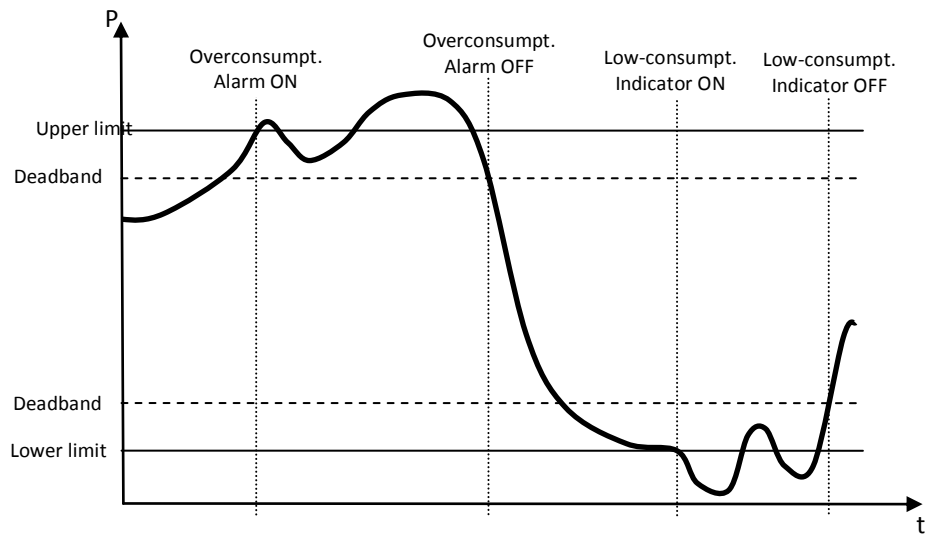


Figure 2.1. Power limit monitoring

These concepts are explained in detail in Section 3.3. *Power limit monitoring.*

3. ETS PARAMETERIZATION

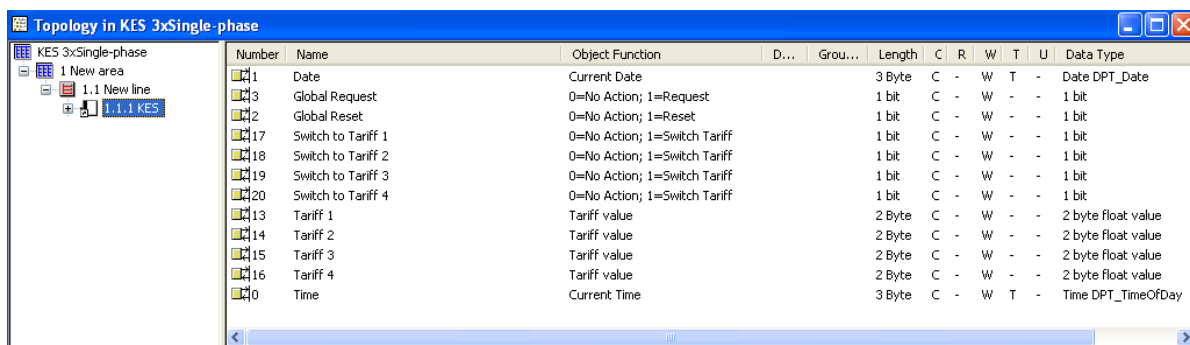
For beginning with the parameterization of KES it is necessary, once the ETS programme has been opened, to import the product database (version 1.0 of the application programme **KES 3xSingle-Phase**).

Next, the device must be added to the project where desired. Right-clicking on the name of the device will display the “Edit parameters” option, which allows the parameterization itself.

In the following sections a detailed explanation about each of the different features of KES under ETS can be found.

3.1. DEFAULT CONFIGURATION

When no particular parameterization has been defined yet, the device and its objects will show as in Figure 3.1.



Number	Name	Object Function	D...	Grou...	Length	C	R	W	T	U	Data Type
1	Date	Current Date			3 Byte	C	-	W	T	-	Date DPT_Date
3	Global Request	0=No Action; 1=Request			1 bit	C	-	W	-	-	1 bit
2	Global Reset	0=No Action; 1=Reset			1 bit	C	-	W	-	-	1 bit
17	Switch to Tariff 1	0=No Action; 1=Switch Tariff			1 bit	C	-	W	-	-	1 bit
18	Switch to Tariff 2	0=No Action; 1=Switch Tariff			1 bit	C	-	W	-	-	1 bit
19	Switch to Tariff 3	0=No Action; 1=Switch Tariff			1 bit	C	-	W	-	-	1 bit
20	Switch to Tariff 4	0=No Action; 1=Switch Tariff			1 bit	C	-	W	-	-	1 bit
13	Tariff 1	Tariff value			2 Byte	C	-	W	-	-	2 byte float value
14	Tariff 2	Tariff value			2 Byte	C	-	W	-	-	2 byte float value
15	Tariff 3	Tariff value			2 Byte	C	-	W	-	-	2 byte float value
16	Tariff 4	Tariff value			2 Byte	C	-	W	-	-	2 byte float value
0	Time	Current Time			3 Byte	C	-	W	T	-	Time DPT_TimeOfDay

Figure 3.1. KES 3xSingle-Phase. Default topology

Objects “Time” and “Date” can be seen in the list. They are intended to set the time and date of the KES device by means of an external KNX clock. Their functionality and everything related to time management will be detailed on *Annex I. Time management*.

“Global Reset” and “Global Request” can be found next. When enabled (by sending the value “1”), the earlier will make every counter and every calculation reset to their default values, so every object related to instant values, estimations, total values and average values that had been parameterized will become zero, as well as every variable needed for calculating. In addition, communication objects related to tariffs and the upper and lower limits will return to their original

value, that is, the one parameterized through the corresponding ETS fields. After that the measuring and calculating process will go on again. The value “0” written to this object will have no effect.

Regarding “Global Request”, the value “1” through this object will make KES send the value of **every** communication object related to consumption (instant values, accumulate values, average values, peak values, etc.) that has been enabled from ETS and, consequently, supposed to be sent by KES at certain times.

The behaviour of the communication objects “Tariff x” and “Switch to tariff x”, shown by default, will be detailed later.

3.2. GENERAL WINDOW

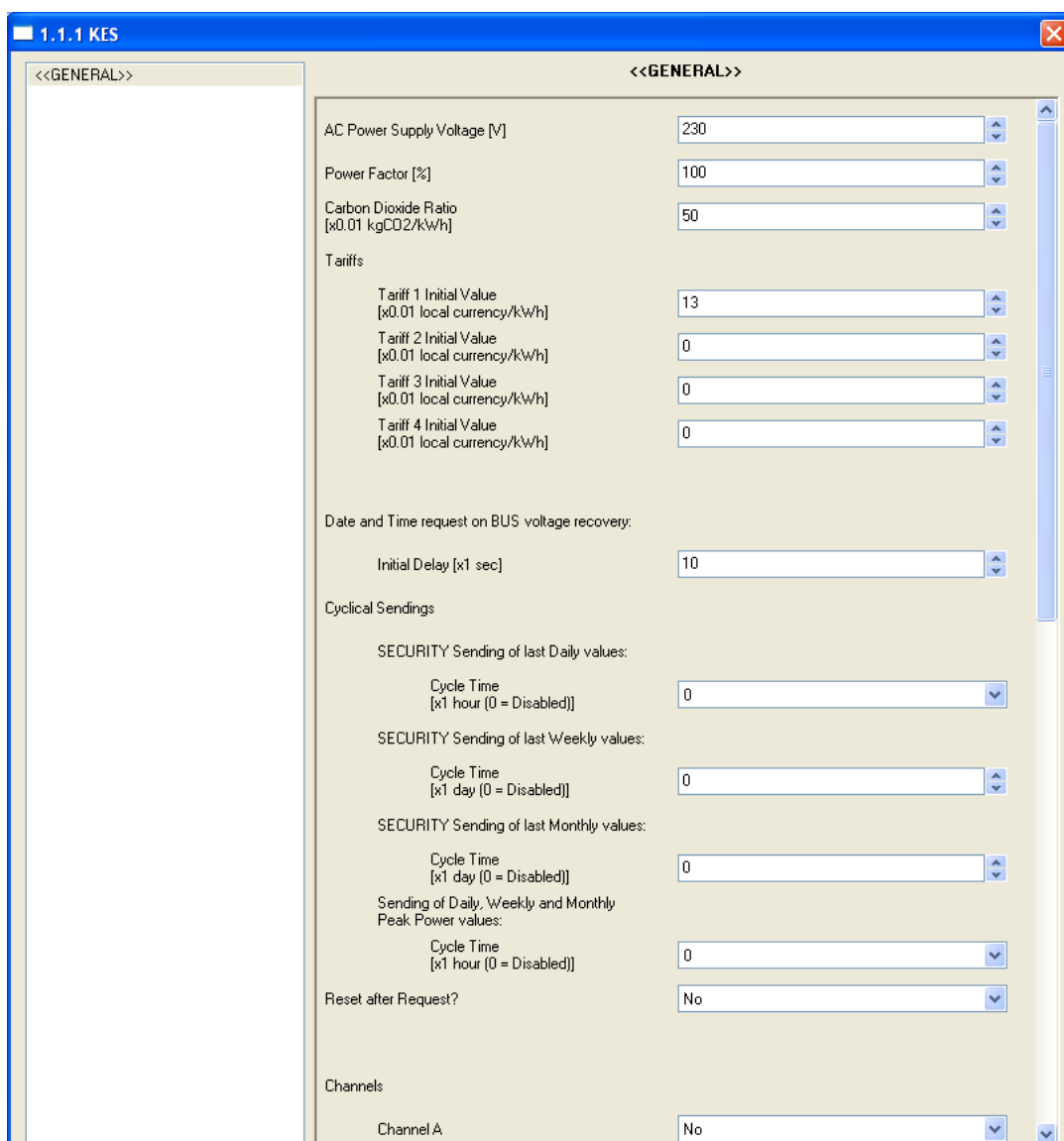





Figure 3.2. Default configuration screen

As seen on Figure 3.2, the **General** screen permits the definition of generic parameters for the operation of the Energy Saver, as well as specifying which channel or channels (A, B, C) will have a ZN1AC-CST60 probe connected for measuring energy consumptions.


Each of these generic parameters is explained next:

 **AC Power Supply Voltage [V]:** this field is to configure the effective voltage, in volts, of the electric network.

 **Power factor [%]:** this parameter defines, as a certain percentage, how the active power (the one actually consumed by the load) and the total supplied power are related.

 **Carbon Dioxide Ratio:** amount of CO₂ (in cents of a Kg) sent to the atmosphere to produce one unit of energy (kWh).


Note: *The value shown by default (0.5 kg CO₂/kWh) is approximately similar to regular ratios in Europe. It is always possible to change it by entering more precise values considering the system to be monitored as well as directly establishing the exact value that can usually be found on the electric bill provided by the electric supplier.*


 **Tariff “x”. Initial value:** initial value for tariff “x” (up to four different tariffs) that will be considered when calculating the cost (estimated, actual or average) of energy consumptions. This cost should be expressed in cents of the local currency.

KES lets define up to four tariffs with a different electric fee each, and switch from one to another in runtime. To make this possible, there are four 1-bit objects (“**Switch to tariff x**”) that allow, by sending the value “1”, changing from one tariff to another for cost calculation purposes. These objects may be linked to an external KNX clock so that, for example, depending on the time band in which KES takes consumption samples, one tariff or another will be used, making cost estimations more precise.

It is also possible to change in runtime the value itself of each tariff. Four 2-byte objects (“**Tariff x**”) can be found for this purpose, making it possible to update the value originally defined by parameter.

Should any of the tariffs be unnecessary, the object “Switch to tariff x” may be left unlinked.

 **Date and time request on bus voltage recovery:** defines how much time (1 to 255 seconds) KES should wait, on voltage recovery, before sending the time and date requests through the KNX bus. See *Annex I. Time management*.

 **Cyclical sendings:** these parameters are destined to enable/disable periodic sendings that affect every active channel. These periodic sendings include both, “security sendings” (in other words, the re-sending of values that had already been calculated and reported to the bus but that are sent again after certain periods) and consecutive sendings of certain updated values. In order to achieve a better understanding of how KES counts these periods and the exact instant when the sending takes place, it is recommended to read *Annex I. Time management*.

➤ **Security sending of last daily values:** allows enabling the periodic sending of values related to the previous day, with the option of selecting cycle times of 1, 2, 3, 4, 6 or 12 hours (choosing “0” disables this sending). This way, all the daily values that had been enabled under “Active energy” (see section 3.4. *Active Energy*) will not only be sent at the end of the day, but also during the following day, every time the selected period expires.

Note: *Regardless of the selected period (0, 1, 2, 4, 6 or 12 hours), the re-sending of the last daily values will not work unless at least one channel (A, B, C) is active and unless at least one daily value has been enabled under the “Active Energy” screen.*

➤ **Security sending of last weekly values:** allows enabling the periodic sending of values related to the previous week, with the option of selecting cycle times of 1 to 7 days (choosing “0” disables this sending). This way, all the weekly values that had been enabled under “Active energy” (see section 3.4. *Active Energy*) will not only be sent at the end of the week, but also during the following week, every time the selected period expires.

Note: *Regardless of the selected period (0-7 days), the re-sending of the last weekly values will not work unless at least one channel (A, B, C) is active and unless at least one weekly value has been enabled under the “Active Energy” screen.*

➤ **Security sending of last monthly values:** allows enabling the periodic sending of values related to the previous month, with the option of selecting cycle times of 1 to 30 days (choosing “0” disables this sending). This way, all the monthly values that had been enabled under “Active energy” (see section 3.4. *Active Energy*) will not only be sent at the end of the month, but also during the following month, every time the selected period expires.

Note: *Regardless of the selected period (0-30 days), the re-sending of the last monthly values will not work unless at least one channel (A, B, C) is active and unless at least one monthly value has been enabled under the “Active Energy” screen.*

➤ **Sending of daily, weekly and monthly peak power values:** allows enabling the periodic sending of the instant value of daily, weekly and monthly peak power, with the option of selecting a cycle time of 1, 2, 3, 4, 6 or 12 hours (choosing “0” disables this sending). This way, the peak power values that had been enabled under “Active Power” (see section 3.5. *Active Power*) will not only be sent when corresponding for each one (at the end of the day, the end of the week or the end of the month) but also every time the selected period expires, letting the bus know how the peak values change during the current day, week and month.

Note: *Regardless of the selected period (0, 1, 2, 3, 4, 6 or 12 hours), the sending of the peak power values will not work unless at least one channel (A, B, C) is active and unless at least one of these peak values has been enabled under the “Active Power” screen.*

🌐 **Reset after Request?:** when enabled (“Yes”), the value of the communication objects and variables related the total consumption in a particular channel (A, B, C) will become zero every time KES responds to a local request for the consumption values of that channel. This local request is ordered by sending the value “1” through the object “[Cx] Request”. (See section 3.3. *Channels* for further information).

🌐 **Channels A, B and C:** allows selecting what channels will be used for measuring. Once enabled (“Yes”), the menu on the left will include a tab for the configuration screen of each active channel (see section 3.3. *Channels*).

🌐 **Logical functions:** when enabling this parameter, the menu on the left will include a tab for the configuration screen of the logical functions. See section 3.6. *Logical functions* for further information.

3.3. CHANNELS

Channel screens will show for the first time as follows:

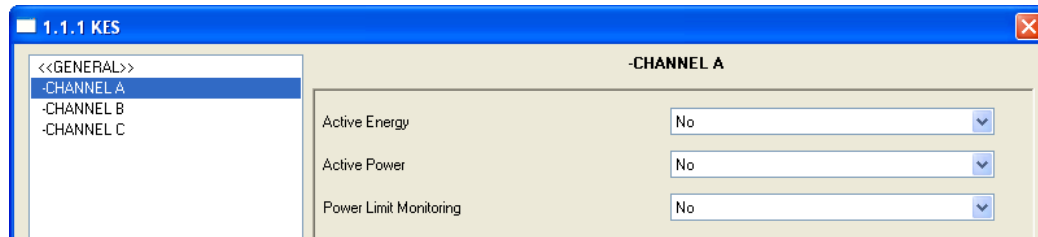


Figure 3.3. Configuration screen for Channels

It is possible to independently enable the parameters **Active Energy**, **Active Power** and **Power Limit Monitoring** for each active channel.

Moreover, after enabling a certain channel, three new 1-bit communication objects will come up per channel: “[Cx] Reset”, “[Cx] Request” and “[Cx] Disable”. Note the distinction between the objects “Global Request” / “Global Reset” (which affect all channels) and the objects “[Cx] Request” and “[Cx] Reset” (which affect only a certain channel).

When the Energy Saver receives the value “1” through the object “[Cx] Reset”, every communication object and variables related to estimations and average / total values of channel “x” will be re-set to zero. Receiving the value “0” will have no effect.

When the Energy Saver receives the value “1” through the object “[Cx] Request”, it will report the accumulate values –since the last reset– for the energy consumption, its cost and the carbon dioxide emissions related to channel “x”. These values are reported through the communication objects “[Cx] Total Energy”, “[Cx] Total Cost” and “[Cx] Total CO2 Emissions”, respectively, which will automatically show up when enabling the corresponding option under ETS. In addition, if the option “Reset after request” has been enabled under the General screen, whenever a “1” is received through the object “[Cx] Request”, a reset process similar to the one described above will also take place. Note that the three total objects are not intended to be read in an independent manner; their values will only be sent to the bus all together, as a response to the “[Cx] Request” object.

The object “[Cx] Disable” allows enabling (when the value “1” is sent) or disabling (when the value “0” is sent) power measures for channel x. When measuring has been disabled, cyclic sendings (those parameterizable from the General screen) as well as the sending of instant values (instant power and peak power) and estimated values (energy and cost) do no longer take place.

Nevertheless, every active energy consumption values (that is hourly, daily, weekly and monthly values) or average values (daily, weekly and monthly) that had been parameterized under ETS will still be sent, although they will not reflect any consumption made while the channel stays disabled.


It is recommended to read the examples from Annex II.

All the parameters from the Active Energy, Active Power and Power Limit Monitoring screens as well as their configurable values for each channel will be detailed in the next pages.

Note: The screen captures shown here may refer to the configuration of Channel B. In that case, similar parameterization and behaviours can be assumed for channels A and C.

3.3.1. ACTIVE ENERGY

This screen allows the configuration of several parameters related to the consumption of active energy (Section 2.2). These show up grouped into sections, as explained below.

 **Estimated values.** This section allows the configuration of parameters regarding the estimation of active energy consumption, expressed in kWh, as well as its estimated cost, in terms of the local currency. This estimation of the active energy consumed is made assuming that the instant power consumption at the moment of the estimation remains constant during the 24 hours of the current day.

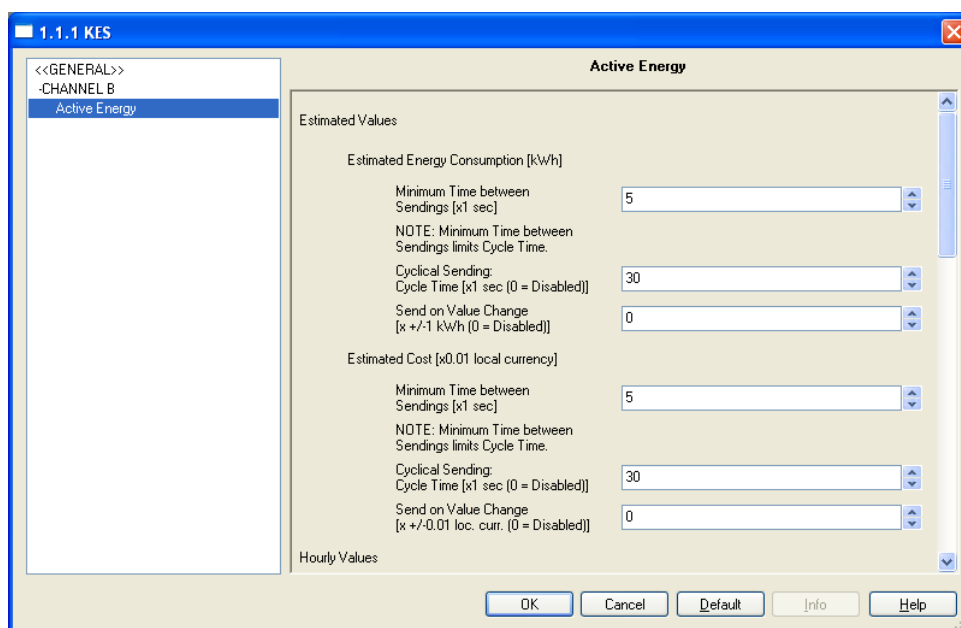



Figure 3.4. Active energy. Estimations.

Related to estimations two communication objects are shown: “[Cx] Estimated energy consumption” and “[Cx] Estimated cost”. The following options are available for them:

- **Minimum time between sendings:** allows setting the minimum time, in seconds, that should pass between two consecutive sendings of the corresponding object (“Estimated energy consumption” and/or “Estimated cost”), no matter if they are cyclic sendings or sendings motivated by a change of the value. Valid times are [5-255] seconds. This minimum time between sendings will only take effect if a value other than 0 is parameterized for fields “Cycle time” or “Send on Value Change”.
- **Cycle time:** allows setting the period for sending the objects “Estimated energy consumption” and/or “Estimated cost”, in seconds. In other words, every how much time these values will be sent to the KNX bus. Note that if the value specified here (x) is lower than the minimum time between sendings (y), the latter will prevail, resulting in sendings taking place only every “y” seconds. If “0” is parameterized for “Cycle time”, no periodic sending of the associated communication object will take place.
- **Send on value change:** allows setting a certain value so that whenever a change in the estimated consumption (both in kWh or in units of the local currency) greater than this value is detected, a new sending of the corresponding communication object (“Estimated energy consumption” or “Estimated cost”) will happen. If there is no need of this feature, just let this field be 0.

Note that if a value other than 0 is set for fields “Cycle time” and “Send on value change”, the associated objects (“Estimated energy consumption” and “Estimated cost”) will also be sent to the KNX bus whenever the value “1” is received through the communication object “Global Request”.

 **Hourly Values.** From this section it is possible to enable the sending of communication objects related to the energy consumption during the last hour, either in kWh (object “[Cx] Last hour energy consumption”) or as the associated cost (object “[Cx] Last hour cost”). These objects will show up once their own parameters have been enabled.

Hourly Values	
Send Last Hour Energy Consumption? [kWh]	Yes
Send Last Hour Cost? [local currency]	Yes

Daily Values. From this section it is possible to enable the sending of communication objects related to the energy consumption (kWh) during the last day, their cost (local currency) and the related CO₂ emissions (kgCO₂). These objects are: “[Cx] Daily Energy Consumption”, “[Cx] Daily Cost” and “[Cx] Daily CO₂ Emissions”. It is also possible to enable the sending of objects representing daily average values: “[Cx] Average Daily Energy Consumption”, “[Cx] Average Daily Cost” and “[Cx] Average Daily CO₂ Emissions”.


Daily Values	
Send Daily Energy Consumption? [kWh]	Yes
Send Daily Cost? [local currency]	Yes
Send Daily CO ₂ Emissions? [kgCO ₂]	Yes
Send Average Daily Energy Consumption? [kWh]	Yes
Send Average Daily Cost? [local currency]	Yes
Daily Average CO ₂ Emissions [kgCO ₂]	Yes

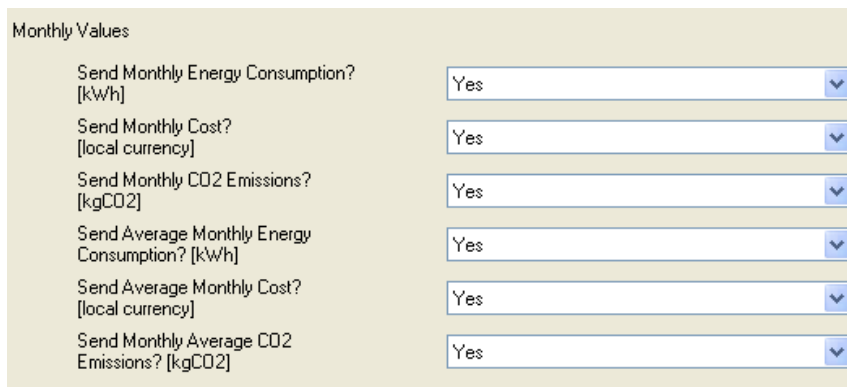
Objects referring to Average Daily values (energy consumption, cost or CO₂ emissions) represent the total consumption (in kWh, local currency or kgCO₂) of the day divided by the number of hours of the day. In other words, they represent the average consumption per hour during the last day.

Weekly Values. The sending of the communication objects related to the energy consumption during the last week, its cost and its required CO₂ emissions can be enabled from here, as well as of the objects representing weekly average values. These are “[Cx] Weekly Energy Consumption”, “[Cx] Weekly Cost”, “[Cx] Weekly CO₂ Emissions”, “[Cx] Average Weekly Energy Consumption”, “[Cx] Average Weekly Cost” and “[Cx] Average Weekly CO₂ Emissions”.

Weekly Values	
Send Weekly Energy Consumption? [kWh]	Yes
Send Weekly Cost? [local currency]	Yes
Send Weekly CO ₂ Emissions? [kgCO ₂]	Yes
Send Average Weekly Energy Consumption? [kWh]	Yes
Send Average Weekly Cost? [local currency]	Yes
Send Average Weekly CO ₂ Emissions? [kgCO ₂]	Yes

Objects referring to Average Weekly values (energy consumption, cost or CO₂ emissions) represent the total consumption (in kWh, local currency or kgCO₂) of the week divided by the number of days of the week. In other words, they represent the average consumption per day during the last week.

 **Monthly Values.** The sending of the communication objects related to the energy consumption during the last month, its cost and its required CO₂ emissions can be enabled from here, as well as the objects representing monthly average values. These are “[Cx] Monthly Energy Consumption”, “[Cx] Monthly Cost”, “[Cx] Monthly CO₂ Emissions”, “[Cx] Average Monthly Energy Consumption”, “[Cx] Average Monthly Cost” and “[Cx] Average Monthly CO₂ Emissions”.




Monthly Values	
Send Monthly Energy Consumption? [kWh]	Yes
Send Monthly Cost? [local currency]	Yes
Send Monthly CO ₂ Emissions? [kgCO ₂]	Yes
Send Average Monthly Energy Consumption? [kWh]	Yes
Send Average Monthly Cost? [local currency]	Yes
Send Monthly Average CO ₂ Emissions? [kgCO ₂]	Yes

Objects referring to Average Monthly values (energy consumption, cost or CO₂ emissions) represent the total consumption (in kWh, local currency or kgCO₂) of the month divided by the number of days of the month. In other words, they represent the average consumption per day during the last month.

3.3.2. ACTIVE POWER

From this screen it is possible to configure several parameters related to the consumption of active power (section 2.2). These show up grouped into different sections, as explained below:


 **Instant power.** This section is to configure those parameters that affect the sending of the communication object “[Cx] Instantaneous Power”.

Instantaneous Power [kW]	
Minimum Time between Sendings [x1 sec]	5
NOTE: Minimum Time between Sendings limits Cycle Time.	
Cyclical Sending: Cycle Time [x1 sec (0 = Disabled)]	0
Send on Value Change [x +/-10 W (0 = Disabled)]	0

These parameters are:

- **Minimum time between sendings:** allows setting the minimum time, in seconds, which should pass between two consecutive sendings of the object “Instant Power”, no matter if they are cyclic sendings or sendings motivated by a change of the value. Valid times are [5-255] seconds. This minimum time between sendings will only take effect if a value other than 0 is parameterized for fields “Cycle time” or “Send on Value Change”.
- **Cycle time:** allows setting the period for sending the object “Instantaneous Power”, in seconds (0 – 255). In other words, every how much time these values will be sent to the KNX bus. Note that if the value specified here (x) is lower than the minimum time between sendings (y), the latter will prevail, resulting in sendings taking place only every “y” seconds. If “0” is parameterized for “Cycle time”, no periodic sending of the associated communication object will take place.
- **Send on value change:** allows setting a certain value so that whenever a change in the instantaneous power greater than that value is detected, a new sending of the corresponding communication object (“Instant Power”) will happen. If there is no need of this feature, just let this field be 0.

Note that if a value other than 0 is set for fields “Cycle time” and “Send on value change”, the object “Instant Power” will also be sent to the KNX bus whenever the value “1” is received through the communication object “Global Request”.

 **Peak Power Values.** This section allows the configuration of the sending of power peaks, that is, the highest instant power values detected during the last hour, day, week or month. Please note that –if enabled– hourly peak values can be sent not only at the end of the last hour but also *before* the end, by means of the parameters **Minimum Time between sendings**, **Cyclical sending** and **Send on value change**, which behave the same way as described above for “Instant power”, making it possible to inform the bus about how the hourly peak value changes throughout the hour. Regarding daily, weekly and monthly peak values, they can be sent not only at the end of the day / week / month, but also periodically (with their

updated instant values, that is, the peak value until the current portion of the current day, of the current week and of the current month), however in this case this must be configured under the General screen, with a unique parameterizable period for the three of them.

The associated objects in this case are “[Cx] Hourly Peak Power”, “[Cx] Daily Peak Power”, “[Cx] Weekly Peak Power” and “[Cx] Monthly Peak Power”

Peak Power Values	
Hourly Peak Power [kW]	
Minimum Time between Sendings [x1 sec]	10
NOTE: Minimum Time between Sendings limits Cycle Time.	
Cyclical Sending: Cycle Time [x1 sec (0 = Disabled)]	0
Send on Value Change [x +10 W (0 = Disabled)]	0
Send Daily Peak Power? [kW]	No
Send Weekly Peak Power? [kW]	No
Send Monthly Peak Power? [kW]	No


3.3.3. POWER LIMIT MONITORING

From this screen it is possible to configure several parameters related to the monitoring of power consumption in relation to upper and lower limits (section 2.3).


Two different sections can be found, one for configuring the parameters of the upper limit and one for configuring the parameters of the lower one.


For the upper limit:


Upper Limit	
Initial Value [x10 W]	0
When over Limit, send...	1
Send Object when under Limit?	No
Send on Value Change [x -10 W (0 = Disabled)]	0

 **Initial value:** allows setting (in tens of a watt) the value that, at least in the beginning, the Energy Saver will assume as the upper limit while measuring power. If the value “0” is set, KES will not consider monitoring power in relation to a particular upper limit. The value of the upper limit, however, can be modified in runtime through the communication object “[Cx]

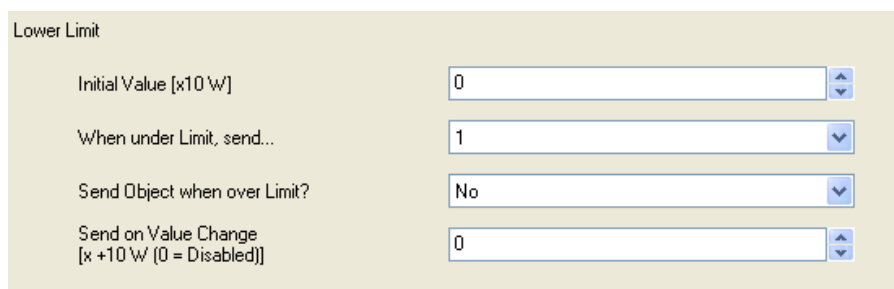
Upper limit”. In the case of parameterizing a value other than 0 for this field, it will be as well necessary to configure the parameters mentioned below, which define the behaviour of the upper limit monitoring.

 **When over limit, send:** allows setting the value (“0” or “1”) that will be written to the object “[Cx] Over-consumption Alarm” when the instant power becomes greater than the value defined for the upper limit, making the user aware of this over-consumption situation in the system.


 **Send Object when under Limit?:** this parameter allows specifying whether the object “[Cx] Over-consumption Alarm” should send or not the inverted value (i.e., “0” if the above parameter is set to “1”) once the instant power is back again under the upper limit (or under the upper limit minus the *deadband* or margin configured through the parameter explained below), making the user aware of the end of the over-consumption situation.

 **Send on value change:** this parameter can be used to avoid consecutive over-consumption alarms every time the instant power –after having exceeded the upper limit and having activated the alarm– is no longer higher than the limit and then exceeds it again. By configuring a certain margin with a value other than 0 (in tens of a watt), every time the over-consumption alarm is triggered, it will be necessary that the measured power decreases to *upper limit minus margin* before further over-consumptions trigger the alarm again. See figure 2.1 (Section 2.3).


For the lower limit:





Lower Limit	
Initial Value [x10 ^W]	0
When under Limit, send...	1
Send Object when over Limit?	No
Send on Value Change [x +10 ^W (0 = Disabled)]	0

 **Initial value:** allows setting (in tens of a watt) the value that, at least in the beginning, the Energy Saver will assume as the lower limit while measuring power. If the value “0” is set, KES will not consider monitoring power in relation to a particular lower limit. The value of the lower limit, however, can be modified in runtime through the communication object “[Cx] Lower limit”. In the case of parameterizing a value other than 0 for this field, it will be as well

necessary to configure the parameters mentioned below, which define the behaviour of the lower limit monitoring.

 **When under limit, send:** allows setting the value (“0” or “1”) that will be written to the object “[Cx] Low consumption Indicator” when the instant power becomes lower than the value defined for the lower limit, making the user aware of this low-consumption situation in the system.

 **Send Object when Over Limit?:** this parameter allows specifying whether the object “[Cx] Low Consumption Indicator” should send or not the inverted value (i.e., “0” if the above parameter is set to “1”) once the instant power is back again over the lower limit (or over the lower limit plus the *deadband* or margin configured through the parameter explained below), making the user aware of the end of the low-consumption situation..

 **Send on value change:** this parameter can be used to avoid consecutive low-consumption indicators every time the instant power –after having become lower than the lower limit and having activated the indicator– is no longer lower than the limit and then becomes lower again. By configuring a certain margin with a value other than 0 (in tens of a watt), every time the low-consumption indicator is triggered, it will be necessary that the measured power reaches *upper limit plus margin* before further low consumptions trigger the indicator again. See figure 2.1 (Section 2.3).

3.4. LOGICAL FUNCTIONS

This functionality in KES is meant to perform binary logic operations with incoming data from the KNX bus, so that the result is then sent through other communication objects specifically enabled in the device for this operation.

Up to 5 different logical functions can be enabled, independent of each other, which can carry out **up to 4 operations** each. To use any of them, it is necessary to enable them in the following ETS window, which appears when selecting “Yes” in the Logical Functions box of the KES General window.

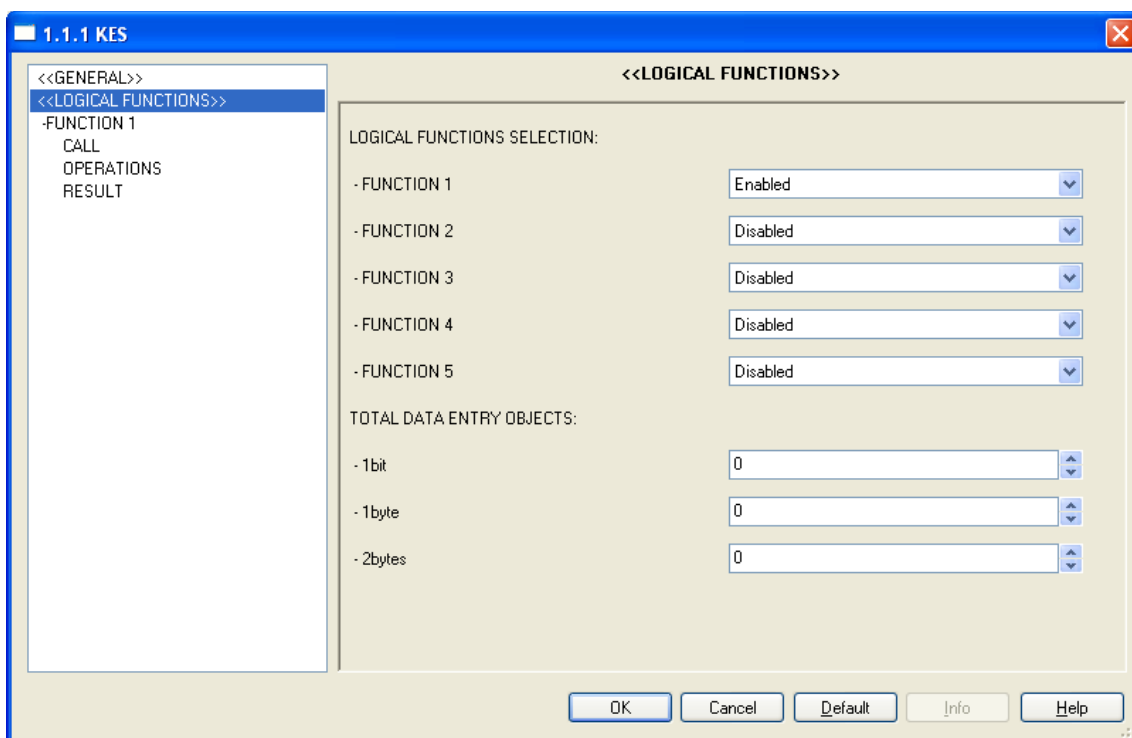


Figure 3.5. Logical functions

To obtain detailed information about the use and ETS parameterization of logical functions, please consult the specific documentation "**Logical Functions X5**", available at <http://www.zennio.com>.

ANNEX I. TIME MANAGEMENT

As seen in Section 3.1, the objects “Date” and “Time” show up by default in ETS when configuring the device. These two objects are aimed at adjusting KES to the date and time of an external KNX clock. However, KES has got its own internal built-in clock, so this two objects are typically only necessary for eventually synchronising the internal clock to the external clock, for example after a parameter download or upon voltage recovery after bus failures (the internal clock only works under bus voltage). Therefore, in usual circumstances it is not necessary that the KNX clock sends these objects whenever time or date change.

Consequently, every time KES is programmed or recovers voltage after a bus failure, these two objects are sent to the bus as a request, with the intention of updating the internal clock according to the time and date that are received as a response from the KNX external clock. If this request receives no response after five seconds, KES will keep retrying by sending both objects up to three more times (with a five-second delay from one to another). If no response is obtained after these four attempts, KES will assume that the current time and date are those that the internal clock had the last time it was working, or –in case KES has been just programmed– that the current date is January 1st 2000 and the current time is 0:00 am.

Regarding cyclic sendings –both the ones enabled from the General screen and the ones enabled from each channel’s own screen–, it is important to remark that all of them will always take place coinciding exactly with hour, day or week changes; never in the middle of two hours or two days or two weeks.

✓ **Example I:**

If one security sending has been defined for every three hours and a parameter download (as well as the consequent time set) takes place at 10:25h am, the first sending will be received at 1:00h pm (that is, three hour changes after the parameter download), not at 1:25h pm.

✓ **Example II:**

If one security sending has been defined for every two days and a parameter download (as well as the consequent time set) takes place on Monday at 10:25h am, the first sending will be received on Wednesday at 0:00h am (that is, two day changes after the parameter download), not at 10:25h am on Wednesday.

In addition, it is important to keep in mind that the timer for periodic sendings is reset at the end of the time unit immediately larger than the one used for defining the sending periods.

✓ **Example III:**

Exactly at midnight, the daily values corresponding to the previous day will always be calculated and sent (concretely, those daily values that had been enabled under ETS). If a security sending of the last daily values has been parameterized for every four hours, the hour counter will be reset every midnight. That's why the first security sending of those values will happen at 4:00h am, no matter if a parameter download (and the consequent time set) had taken place at 10:50h pm.

✓ **Example IV:**

Exactly every Sunday at midnight (i.e., in the exact moment the day changes into Monday) the weekly values corresponding to the previous week will always be calculated and sent (concretely, those weekly values that had been enabled under ETS). If a security sending of the last weekly values has been parameterized for every four days, the day counter will be reset with this week change. This way, the next security re-sending will happen at the end of next Thursday, that is, four day changes after the values were calculated. Considering that the number of days in a week (7) is not a multiple of 4, this behaviour allows that the security sendings take place the same days every week (in our case, when switching from Thursday to Friday, apart from the ordinary sending at the end of Sunday), instead of spreading a certain delay from week to week.

On the other hand, it must also be remarked that bus failures will also reset the timing for the periodical sendings, which is anyway conditioned to the behaviour described in the two previous examples.

✓ **Example V:**

A security sending is parameterized for every six hours and a parameter download takes place at 9:25h am, after which the internal clock is set to current time. When it's 3:00h pm, the first periodical sending happens. After that, at 5:15h pm, a bus failure takes place (or, alternatively, the device is turned off for a few minutes). At 5:25h pm the voltage is recovered and the internal clock is set again to current time (otherwise, it would assume that it is still 5:15h pm). Since the voltage failure has caused a timing reset of the periodical sendings, the next security sending will happen at 11:00h pm, even if the previous one had taken place at 3:00h pm.

Finally, and with the hope of allowing a better comprehension of the behaviour of KES under hour changes (such as summer to winter & winter to summer hour corrections) or under consecutive time sets with testing purposes, it is important to note that the timers in KES do not really count *the number of hours that go by*, but *the number of forward hour changes*. Of course, all considerations that have already made will also apply here, as shown with the following examples:

✓ **Example VI:**

A periodic sending is parameterized for every three hours and a parameter download takes place at 9:25h am, after which the time of the internal clock of KES is set. Five minutes after, the clock is again set, in this case to 10:15h am. A few seconds later, it is set to 1:38h pm. And a little later, it is set to 5:06h pm. Immediately after this time change, the device will execute the first periodic sending. The reason is that, according to KES, three forward hour changes have taken place since it began working: the first one, from 9h to 10h; the second one, from 10h to 1h; and the third one, from 1h to 5h. Note that if no more fictitious time changes are forced after setting the clock to 5:06h, the device will execute the second security sending when its internal clock shows 8:00h pm. Moreover, as said before, after midnight the periodic timer will be set to zero, so the first sending of the next day will take place at 3:00h am.

✓ **Example VII:**

A periodic sending has been parameterized for every three hours. The device has been turned on and the internal clock has been correctly set. The night when clocks must be put one hour back, as soon as it is 3:00h am the periodic sending takes place, and after that the external KNX clock sends an order to the internal clock so that the time is set back to 2:00h am. This backtracking does not affect the timer of the periodic sendings, however, once it is 3:00h am again, the device does consider that a new forward hour change has taken place (as it has actually been). Because of this, the next periodic sending will take place at 5:00h am (which would be 6:00h am if no backtracking had happened). Finally, after the next midnight, the timers will be reset again, so during the next days there won't be sendings at 5:00h am, but at 6:00h am.

ANNEX II. FURTHER EXAMPLES

With the aim of making the comprehension of how KES works simpler and of presenting the multiple parameterizable communication objects in their context, a few more examples will be presented next. The intention is that they cover exhaustively enough most of the doubts that may rise during the initial contact with the device. Of course, these examples are complemented with those already included in *Annex I. Time Management*.

Note: *A few seconds of sampling are always needed by KES before most of the objects and variables are updated. If for testing / simulation purposes several hour changes are forced with less than 5 seconds between each and the next one, KES may report incorrect or null values through the objects sent to the bus (such as “Last hour Energy Consumption”).*

✓ Example 1: total consumption

The simplest usage of KES Energy Saver consists in destining it to calculate total consumptions, so that they can be contrasted with those from the electric bill. For that, it is enough with parameterizing the following:

- 🔴 **General screen:** only one tariff, with a cost of 15 cents per kWh, and a CO₂ ratio of 50. Voltage, as corresponding to the system. Power factor, 100. Enable the corresponding channel. All cyclic sendings, left disabled.
- 🔴 **Active Energy screen:** values 5, 0 and 0 for section “Estimated energy consumption”. Values 5, 0 and 0 for section “Estimated cost”. Parameters “Daily energy consumption”, “Daily cost”, “Monthly energy consumption” and “Monthly cost”, enabled. All other parameters, disabled.
- 🔴 **Active Power screen:** values 5, 0 and 0 for section “Instantaneous Power” and 10, 0 and 0 for “Hourly Peak Power”. All other parameters, disabled.

In such circumstances, the only functions KES will develop are:

- 1) Sending, every midnight, the objects “Daily energy consumption” and “Daily cost”, expressed in kWh and monetary units, respectively.

- 2) Sending, the 1st day of every month at 0:00h am, the objects "Monthly energy consumption" and "Monthly cost", expressed in kWh and monetary units respectively.

For simplicity purposes, the probe can be supposed to be measuring the consumption of a 25-watt bulb, and a certain month can be supposed with 22 workable days –during which the bulb is switched on between 8h in the morning and 6h in the afternoon– and 8 non-workable days –during which the bulb remains off–. In such case:

- 1) Every midnight, the object "Dialy energy consumption" sent to the bus will be 0 when the previous day has been non-workable, or $(8h * 0.025 \text{ kW}) = 0.2 \text{ kWh}$ if workable, while the object "Daily cost" will have a value of 0 after a non-workable day and of $(0.2 \text{ kWh} * 0.15 \text{ m.u.}) = 0.03 \text{ monetary units}$ after any other day.
- 2) After every last day of a month, at midnight, the object "Monthly energy consumption" sent to the bus will have a value of $(0 \text{ kWh} * 8 \text{ days}) + (0.2 \text{ kWh} * 22 \text{ days}) = 4.4 \text{ kWh}$, while the object "Monthly cost" will be $(4.4 \text{ kWh} * 0.15) = 0.66 \text{ monetary units}$.

This way, after every month KES will automatically report in what grade the bulb contributed to the consumption and the cost of the electric bill.

Alternatively, if desired, periodic security sendings can be enabled from General screen, which allows re-sending the last daily and/or monthly values, so that not only are they sent everyday at midnight or every 1st day of a month (respectively), but also successively during later periods.

Last, there is always the possibility of making global or local requests (through the objects "Global request" and "[Cx] Request") at the end of the month so that KES reports the value of "Total cost" and "Total energy". This procedure will be quite similar to reading the ordinary counter installed by the power supplier, as it is enough to calculate the difference between the current value of "Total energy" and that from last month in order to obtain the monthly consumption (in the case of this example, the consumption made by the bulb).

Note: *Objects related to energy consumption always report their values (although objects related to cost or CO₂ emissions do not) rounded to the nearest integer. That's why the 0.2 kWh mentioned in the above example may show up as 0 kWh, although the 0.2 kWh are actually considered for further calculation.*

✓ Example 2: disabled channel

Four different cases are now presented in relation to how KES works when the channel where measures are being made is disabled:

 **CASE 1**



Let a power consumption of 3 kW happen during 24h. If the channel is permanently enabled, at the end of the day:

[Cx] Daily Energy Consumption = 72 kWh

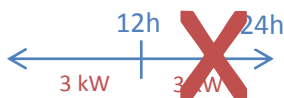
[Cx] Average Daily Energy Consumption = $72 / 24 = 3$ kWh per hour

If during the rest of the week the channel is left enabled and the power consumption keeps being 3 kW:

[Cx] Weekly energy consumption = 504 kWh

[Cx] Average Weekly Energy Consumption = $504 / 7 = 72$ kWh per day

 **CASE 2**



Similar case, but one of the days the channel is left disabled during 12 hours (no matter which ones). In such case, that day:

[Cx] Daily Energy Consumption = 36 kWh

[Cx] Average Daily Energy Consumption = $36 / 12 = 3$ kWh

If the channel is left enabled for the rest of the week and the power consumption is still 3 kW, when the week ends:

[Cx] Weekly Energy Consumption = 468 kWh

[Cx] Average Weekly Energy Consumption = $468 / 7 = 66.85$ kWh ≈ 67 kWh

 **CASE 3**



Similar situation, but one of the days the channel is left disabled during the whole day. That day:

[Cx] Daily Energy Consumption = 0 kWh

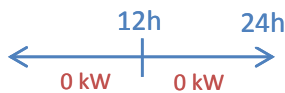
[Cx] Average Daily Energy Consumption = $0 / 0 = 0$ kWh per hour

If the channel is left enabled for the rest of the week and the power consumption is still 3 kW, when the week ends, that day is not taken into account:

[Cx] Weekly Energy Consumption = 432 kWh

[Cx] Average Weekly Energy Consumption = $432 / 6 = 72$ kWh per kWh

CASE 4



In the case of a total consumption of exactly zero at the end of the day (note that a null consumption is really a very remote situation), if the channel has stayed enabled for the whole day, then that day will be taken into account for weekly values. For example, suppose a daily consumption of 72 kWh every other day of the week:

[Cx] Daily Energy Consumption = 0 kWh

[Cx] Average Daily Energy Consumption = $0 / 24 = 0$ kWh per hour

[Cx] Weekly Energy Consumption = 432 kWh

[Cx] Average Weekly Energy Consumption = $432 / 7 \approx 62$ kWh per day

CASE 5



A particular case is that of having, during one day, a total consumption of zero (note that such situation is quite improbable) and the channel disabled for a few hours. In that case, for weekly calculation, that day will only be taken into account if during the day change the channel is found active.

If the channel is only disabled that day from 7h am to 1h pm:

[Cx] Daily Energy Consumption = 0 kWh

[Cx] Average Daily Energy Consumption = $0 / 18 = 0$ kWh

[Cx] Weekly Energy Consumption = 432 kWh

[Cx] Average Weekly Energy Consumption = $432 / 7 \approx 62$ kWh

If the channel is only disabled that day from 7h pm to 12h pm:

[Cx] Daily Energy Consumption = 0 kWh

[Cx] Average Daily Energy Consumption = $0 / 19 = 0$ kWh

[Cx] Weekly Energy Consumption = 432 kWh

[Cx] Average Weekly Energy Consumption = $432 / 6 = 72$ kWh

(For the other six days of the week, assume the channel is permanently enabled and a daily consumption of 72 kWh).

✓ Example 3: disabled channel

While measuring the electric consumption of a certain device that remains always turned on and that consumes 200 W constantly, we leave enabled the corresponding channel of KES between 0:00h am and 11:00h am on Monday. Exactly when it's 11:00h am, we deactivate the channel through the object "[Cx] Disable". When it turns midnight, KES will consider that the total consumption during the day has been 2.2 kWh (that is, 200 watts constantly during 11 hours, since there was no measuring between 11:00h am and midnight) and that the average daily consumption has been 200 watts per hour (2.2 kWh / 11 hours).

Exactly after midnight, the channel is enabled again and remains this way indefinitely, measuring a 200 Wh consumption every hour. When the week is over (after midnight from Sunday to Monday) the earlier Monday will be considered as a regular day in order to calculate the average weekly consumption per day. In other words:

- **Weekly consumption** = 2.2 kWh + (6 days x (0.2 kW x 24 hour)) = 31 kWh.
- **Average weekly consumption** = 31 kWh / 7 days = 4.43 kWh per day.

Let the device whose consumption we are measuring be now turned on only on Tuesday, Thursday and Saturday and let the channel be always enabled. In that case, there will be a consumption of 0 kWh on Monday, Wednesday, Friday and Sunday. The total consumption for Tuesday, Thursday and Saturday will be (3 days x (0.2 kW x 24 hours)) = 14.4 kWh. However, once the week is over, the average weekly consumption will be 14.4 kWh / 7 days = 2.06 kWh. The division is by 7 because the channel stayed enabled during seven day forward changes, even the days in which there was a consumption of exactly zero.

✓ Example 4: disabled channel

Suppose the energy consumption of an office is going to be measured. Most of everyday consumption takes place during the 8 working hours, and although there is not a consumption of exactly 0 kWh during the rest of the day, it is not going to be measured. To achieve this, a case similar to #2 in Example 2 must be considered. Enabling the channel only during 8 hours a day will cause that any consumption made during the other 16 hours is discarded, and that the average daily consumption is obtained dividing by 8 (not 24). Regarding weekly and monthly values, only those days in which not only there was a null total consumption but also the channel was detected disabled at the end of the day (midnight) will not be considered.

✓ Example 5: tariffs

The following tariff parameterization is applied to KES:


- Initial value for Tariff 1: 15 cents
- Initial value for Tariff 2: 16 cents
- Initial value for Tariff 3: 0 cents
- Initial value for Tariff 4: 0 cents

Assume an instant power consumption of 2 kW and that KES Energy Saver is turned on at midnight.

- 1) The Estimated cost will be: $2 \text{ kW} * 24\text{h} * 0.15 \text{ m.u.} = 7.2$ monetary units.
- 2) When it's 1:00h, Last hour cost will be $2 \text{ kW} * 1\text{h} * 0.15 \text{ m.u.} = 0.30$ m.u.
- 3) At 1:22h, if a 1 is sent through "Switch to tariff 2", the Estimated cost will become $2 \text{ kW} * 24\text{h} * 0.16 \text{ m.u.} = 7.68$ monetary units.
- 4) At 2:00h, Last hour cost results to be $2 \text{ kW} * 1\text{h} * 0.16 \text{ m.u.} = 0.32$ m.u.
(As seen, the hourly cost has been rounded because of applying only the new tariff).
- 5) At 2:33h, a 1 is sent through object "Switch to tariff 3".
- 6) When it's 3:00h, Last hour cost is $2 \text{ kW} * 1\text{h} * 0 \text{ m.u.} = 0$ m.u.
(As seen, setting the initial value of a tariff as zero doesn't disable it; for effectively disabling it, it is necessary to leave the object "Switch to tariff x" unlinked).
- 7) At 3:14h, the value 0.20 (m.u.) is sent through object "Tariff 3".
- 8) When it's 4:00h, Last hour cost will be $2 \text{ kW} * 1\text{h} * 0.20 \text{ m.u.} = 0.4$ m.u.

✓ Example 6: power factor

Assume a system where the power factor is 90%, i.e., where 10% of the effective energy (that supplied by the electric company) is not actually converted into active (or useful) energy, that is, consumed by the devices.

-  If a power factor of 90% is parameterized, the values sent by KES to the bus in relation to power and energy will correspond to those actually consumed by the system, although the amount of effective energy that is really transmitted by the line is a bit higher.

- If a power factor of 100% is parameterized, KES will consider the total energy detected in the line, without applying the pertinent corrections to discard the portion of energy that is not actually consumed.
- A parameterization with a power factor of 80% will cause excessive corrections made by KES, which will send values to the bus that are smaller than the effective ones and also than the active ones.

✓ Example 7: power limits

Assume a Power Limit parameterization as follows:

• Upper limit:

Initial value = 200 (tens of a watt)

When over limit, send = 1

Send object when under limit = Yes

Send on value change = 20 (tens of a watt)

• Lower limit:

Initial value = 100 (tens of a watt)

When under limit, send = 1

Send object when over limit? = Yes

Send on value change = 20 (tens of a watt)

In such case, the following can be tested:

- 1) As soon as the instant power is over 2 kW, a “1” will be sent through object “[Cx] Over-consumption alarm”.
- 2) If the value goes down to 1.92 kW, nothing will happen. Nothing will happen either if it goes over 2 kW again.
- 3) If the value decreases to 1.78 kW, the inverted value (“0”) will be sent through object “[Cx] Over-consumption alarm”. If after that it increases again to over 2 kW, one “1” will be sent through the same object.
- 4) If the value now goes to 0.95 kW, one “0” will be first sent through “[Cx] Over-consumption alarm” and, afterwards, one “1” through “[Cx] Low consumption indicator”.
- 5) A new value of 1.05 kW will not have an effect, as well as a later decrease to 0.95 kW.


- 6) If the instant power goes to 1.25 kW, the inverted value (a “0”) will be sent through “[Cx] Low consumption indicator”. If it goes then under 1 kW, one “1” will be sent through the same object.
- 7) Finally, if the values “3.0” and “4.0” are sent through objects “[Cx] Lower limit” and “[Cx] Upper limit” respectively, it will become necessary that the power goes over 3.20 kW before a “0” is sent through “[Cx] Low consumption indicator”, and that it goes over 4 kW before a “1” is sent through “[Cx] Over-consumption alarm”.


Note: *Special care must be taken when writing to objects “[Cx] Upper limit” and “[Cx] Lower limit”. Values which are too close (enough to have a difference smaller than the defined margin bands) or that do not satisfy the condition $L_{upper} > L_{lower}$ will cause behaviours with no logical sense.*


✓ Example 8: global test

This example, eminently practical, pretends simulating in only a few minutes how KES would work while time passes (several hours, several days and several weeks). At the end of the example a small graph has been included to show the hour, day and week changes that are being simulated.

To begin with, the following ETS parameterization is proposed for KES:

 **General screen:** every cyclic sending and security sending enabled with the most frequent option available (every hour or every day, depending on the case). Only one tariff, with a cost of 5000 (cents of the local currency) per kWh and a CO₂ ratio of 50. As the voltage, the one corresponding to the system. And the power factor, 100.

 **Active Energy screen:** values 5, 30 and 0 for section “Estimated energy consumption”. Values 5, 30 and 0 for section “Estimated cost”. All the other parameters enabled.

 **Active Power screen:** values 5, 30 and 0 for section “Instant power”. Values 10, 30 and 0 for section Hourly peak power. All the other parameters, enabled.

After downloading the parameters, the internal clock should be set to 9:00h am of the first Monday of the current month.

1) Connect, for example, a 2000 W load to the corresponding channel:

- After 30 seconds after the time set, the object Instantaneous power will be sent to the bus, with a value of 2 kW.
 - Estimated energy consumption \approx 48 kWh.
This is calculated as the product of 2 KW and 24 hours, which indicates the estimated daily consumption assuming the current instant power as constant.
 - Estimated cost \approx 2400 monetary units.
This is the product of 48 kWh and the parameterized 50 monetary units.
 - Hourly Peak Power \approx 2 kW
This is the highest value of the instant power detected till now since the time set.
- 2) After 5 minutes since 1), replace the 2000 W load with a 3000 W load and after a few seconds disconnect it, letting the consumption be null.**
- The next Hourly Peak Power (which is sent every 30 seconds) will be 3 kW.
- 3) After 5 minutes since 2), connect back the 2000 W load and set the internal clock to 10:00h am.**
- After the time change, the object Last hour energy consumption will be sent, containing a value of 1 kWh.

According to the device, the last hour only lasted for ten minutes, during five of which there was a consumption of 2 kW and during the other five of which there was a consumption of 0 kW. Therefore, KES estimates that between 9:00h and 10:00h there should have been, proportionally, a consumption of 1 kWh.
 - Last hour cost \approx 50 monetary units.
50 is the cost corresponding to the kilowatt-hour that was estimated by the device for the period 9:00h to 10:00h.
- 4) After some minutes since 3), set the internal clock to next Tuesday (any time would be OK), maintaining the 2000 W load.**
- Daily energy consumption = 3.
According to the device, last Monday only lasted for two hours (each of which lasted itself for only a few minutes, although not the same amount in both cases), the first one with a consumption of 1 kWh and the second one with a consumption

of 2 kWh. That is why it estimates that the total consumption for Monday was 1 kWh + 2 kWh = 3 kWh.

- Daily cost \approx 150 monetary units.
 $3 \text{ kWh} * 50 \text{ m.u.} = 150 \text{ m.u.}$
- Daily CO2 emissions \approx 1.5 kg
Obtained as $(50 \text{ kg} * 0.01) * 3 \text{ kWh}$.
- Daily Peak Power \approx 3 kW.
This is the highest value detected for the instant power during the previous day.
- Average daily energy consumption = 2 kWh.
This value represents the energy amount that, as an average, is consumed per hour during the day. Considering that objects related to energetic consumption are always rounded to the closest integer, this “2” is consistent with the result of dividing the 3 kWh consumed on Monday by the number of hours of that day (two).
- Average daily CO2 emissions \approx 0.75.
This value is the result of $(50 \text{ kg} * 0.01) * 3 \text{ kWh} / 2 \text{ hours}$.
- Average daily cost \approx 75 monetary units.
This object indicates the cost per hour that, as an average, will be paid for the energy consumption of Monday. The value sent through this object is obtained with the following equation: $3 \text{ kWh} * 50 \text{ monetary units} / 2 \text{ hour}$.

5) After some minutes since 4), set the internal clock to next Monday (any time would be OK), maintaining the 2000 W load.

- Weekly energy consumption = 5.
According to the device, last week only lasted for two days (with a different amount of hours each, and with a different length each of these hours). That is why KES estimates that the total consumption for the week was 3 kWh from Monday and 2 kWh from Tuesday.
- Weekly cost \approx 250 monetary units.
 $5 \text{ kWh} * 50 \text{ m.u.} = 250 \text{ m.u.}$
- Weekly CO2 emissions \approx 2.5 kg.

Obtained as $(50 \text{ kg} * 0.01) * 5 \text{ kWh}$.

- Average daily energy consumption = 2 kWh.
According to the device, the previous day lasted only for one hour (with a consumption of 2 kWh). Therefore, the average daily consumption was $2 \text{ kWh} / 1 \text{ hour} = 2 \text{ kWh per hour}$.
- Average daily CO2 emissions $\approx 0.5 \text{ kg}$.
This value is the result of calculating $(50 \text{ kg} * 0.01) * 2 \text{ kWh} / 1 \text{ hour}$.
- Average daily cost ≈ 100 monetary units.
Obtained by multiplying 2 kWh by 50 monetary units, and dividing this by the only hour of the previous day.
- Weekly Peak Power $\approx 3 \text{ kW}$.
- Average weekly energy consumption = 3 kWh
Obtained by rounding to the nearest integer value the result of dividing the 5 kWh consumed during the previous week by the number of days of the week.
- Average weekly CO2 emissions ≈ 1.25 .
This value comes from operating $(50 \text{ kg} * 0.01) * 5 \text{ kWh} / 2 \text{ days}$.
- Average weekly cost ≈ 125 monetary units.
5 kWh were consumed at 50 monetary units per kWh, which is then divided by the two days of the previous week.

6) After some minutes since 5), set the internal clock to next Monday (any time would be OK), maintaining the 2000 W load.

- Since the previous week contained only one day, the objects Daily energy consumption and Weekly energy consumption must have the same value, 2 kWh in this case.
- Average weekly energy consumption = 2 kWh per day.
 $2 \text{ kWh in total} / 1 \text{ day} = 2 \text{ kWh per day}$.
- Average weekly CO2 emissions $\approx 1 \text{ kg per day}$.
 $(50 \text{ kg} * 0.01) * 2 \text{ kWh} / 1 \text{ day} = 1$.

- Average weekly cost ≈ 100 monetary units per day.
 $2 \text{ kWh} * 50 \text{ monetary units} / 1 \text{ day} = 100 \text{ m.u.}$
- Weekly Peak Power $\approx 2 \text{ kW}$.

7) After some minutes since 6), set the internal clock to the first day of the next month, maintaining the 2000 W load.

- Monthly energy consumption = 9 kWh.
 Obtained by summing up (and rounding to the nearest integer value) 5 kWh from the first week of the previous month, plus 2 kWh from the second week, plus 2 kWh from the third one.
- Monthly cost ≈ 450 monetary units.
 Obtained by multiplying 9 kWh by 50 monetary units.
- Monthly CO2 emissions $\approx 4.5 \text{ kg}$.
 In other words, $(50 \text{ kg} * 0.01) * 9 \text{ kWh}$.
- Monthly Peak Power $\approx 3 \text{ kWh}$.
 This value is a consequence of the few seconds the 3000 W load was connected.
- Average monthly energy consumption = 2 kWh per day.
 2 is the nearest integer to the result of dividing 9 kWh by the number of days (4) of the previous month.
- Average monthly cost ≈ 112.5 monetary units per day.
 $50 \text{ u.m} * 9 \text{ kWh} / 4 \text{ days} = 112.5 \text{ m.u.}$
- Average monthly CO2 emissions $\approx 1.125 \text{ kg}$ per day.
 Obtained as $(50 \text{ kg} * 0.01) * 9 \text{ kWh} / 4 \text{ days}$.

8) After some minutes since 7), set the internal clock to the first day of the following month (that is, two months from now on), maintaining the same load.

- Monthly Peak Power $\approx 2 \text{ kW}$.
- Average monthly energy consumption = 2 kWh per day.
 The previous month consisted of only one hour, with a consumption of 2 kWh.
 This total consumption is divided by only one day.

- Average monthly cost ≈ 100 m.u. per day.
Those 2 kWh consumed throughout the month have an associated cost of 50 u.m each. Then, we divide by only one day.
- Average monthly CO2 emissions ≈ 1 kg. per day.
 $(50 \times 0.01) * 2 \text{ kWh} / 1 \text{ day} = 1$.

Simulation scheme

Month 1

WEEK 1

- **Monday** $\begin{array}{c} 2 \text{ kW} \quad 0 \text{ kW} \quad 2 \text{ kW} \\ |-----|-----|-----| \\ \quad 5' \quad \quad 5' \quad \quad 5' \end{array}$ **Monday consumption = 1 kWh + 2 kWh = 3 kWh**
- **Tuesday** $\begin{array}{c} 2 \text{ kW} \\ |-----| \end{array}$ **Tuesday consumption = 2 kWh**

WEEK 2

- **Monday** $\begin{array}{c} 2 \text{ kW} \\ |-----| \end{array}$ **Monday consumption = 2 kWh**

WEEK 3

- **Monday** $\begin{array}{c} 2 \text{ kW} \\ |-----| \end{array}$ **Monday consumption = 2 kWh**

Monthly energy consumption = 9 kWh.

Average monthly energy consumption = 9 kWh / 4 days = 2.25 kWh per day

Month 2

WEEK 1

- **1st day** $\begin{array}{c} 2 \text{ kW} \\ |-----| \end{array}$ **Day consumption = 2 kWh**

Monthly energy consumption = 2 kWh.

Average monthly energy consumption = 2 kWh / 1 day = 2 kWh per day

ANNEX III. COMMUNICATION OBJECTS

SECTION	NUMBER	SIZE	IN/OUT	FLAGS	VALUES			NAME	DESCRIPTION
					RANGE	1st TIME	RESET		
GENERAL OBJECTS	0	3 bytes	I	WT	0:00:00 - 23:59:59	0:00:00	Previous	Time	Current Time
	1	3 bytes	I	WT	01/01/1990 – 31/12/2089	01/01/2000	Previous	Date	Current Date
	2	1 bit	I	W	0/1	0	0	Global Reset	0=No action; 1=Reset
	3	1 bit	I	W	0/1	0	0	Global Request	0=No action; 1=Request
	13-16	2 bytes	I	W	0-65535	Depends on initial paramet.	Previous	Tariff x [x=1...4]	Tariff value
	17-20	1 bit	I	W	0/1	0	0	Switch to tariff x [x=1...4]	0=No action; 1=Switch Tariff
CHANNELS: GENERAL OBJECTS	4-6	1 bit	I	W	0/1	0	0	[Cx] Reset	0=No action; 1=Reset
	7-9	1 bit	I	W	0/1	0	0	[Cx] Request	0=No action; 1=Request
	10-12	1 bit	I	W	0/1	0	Previous	[Cx] Disable	0=Disable; 1=Enable
CHANNELS: ACTIVE ENERGY OBJECTS	21-23	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Estimated energy consumption	Estimated energy consumption
	24-26	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Estimated cost	Estimated cost
	27-29	2 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Last hour energy consumption	Last hour energy consumption

SECTION	NUMBER	SIZE	IN/OUT	FLAGS	VALUES			NAME	DESCRIPTION
					RANGE	1st TIME	RESET		
CHANNELS: ACTIVE ENERGY OBJECTS	30-32	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Last hour cost	Last hour cost
	33-35	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Daily energy consumption	Daily energy consumption
	36-38	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Daily cost	Daily cost
	39-41	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Daily CO2 emissions	Daily CO2 emissions
	42-44	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Average daily energy consumption	Average daily energy consumption
	45-47	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Average daily cost	Average daily cost
	48-50	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Average daily CO2 emissions	Average daily CO2 emissions
	51-53	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Weekly energy consumption	Weekly energy consumption
	54-56	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Weekly cost	Weekly cost
	57-59	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Weekly CO2 emissions	Weekly CO2 emissions
	60-62	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Average weekly energy consumption	Average weekly energy consumption
	63-65	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Average weekly cost	Average weekly cost
	66-68	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Average weekly CO2 emissions	Average weekly CO2 emissions
	69-71	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Monthly energy consumption	Monthly energy consumption
72-74	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Monthly cost	Monthly cost	

SECTION	NUMBER	SIZE	IN/OUT	FLAGS	VALUES			NAME	DESCRIPTION
					RANGE	1st TIME	RESET		
CHANNELS: ACTIVE ENERGY OBJECTS	75-77	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Monthly CO2 emissions	Monthly CO2 emissions
	78-80	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Average monthly energy consumption	Average monthly energy consumption
	81-83	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Average monthly cost	Average monthly cost
	84-86	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Average monthly CO2 emissions	Average monthly CO2 emissions
	87-89	4 bytes	O	RT	0 - 2 ³²	0	Previous	[Cx] Total energy	Total energy consumption
	90-92	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Total cost	Total cost
	93-95	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Total CO2 emissions	Total CO2 emissions
CHANNELS: ACTIVE POWER OBJECTS	96-98	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Instantaneous Power	Instantaneous power
	99-101	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Hourly Peak Power	Hourly Peak Power
	102-104	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Daily Peak Power	Daily Peak Power
	105-107	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Weekly Peak Power	Weekly Peak Power
	108-110	2 bytes	O	RT	0 - 670760	0	Previous	[Cx] Monthly Peak Power	Monthly Peak Power
CHANNELS: POWER LIMIT OBJECTS	111-113	2 bytes	I	W	0 - 670760	Depends on initial paramet.	Previous	[Cx] Upper limit	Upper limit value
	114-116	1 bit	O	RT	0/1	Disabled ("0" or "1")	Previous	[Cx] Over-consumption alarm	Alarm -> Sending of "0" or "1"
	117-119	2 bytes	I	W	0 - 670760	Depends on initial paramet.	Previous	[Cx] Lower limit	Lower limit value
	120-122	1 bit	O	RT	0/1	Disabled ("0" o "1")	Previous	[Cx] Low consumption indicator	Ind. -> Sending of "0" or "1"

SECTION	NUMBER	SIZE	IN/OUT	FLAGS	VALUES			NAME	DESCRIPTION
					RANGE	1st TIME	RESET		
LOGICAL FUNCTIONS	123-138	1 bit	I	W	0/1	0	Previous	[FL] Data Entry (1bit) 1 ... [FL] Data Entry (1bit) 16	Binary data entry (0/1) ... Binary data entry (0/1)
	139-146	1 byte	I	W	0-255	0	Previous	[FL] Data Entry (1byte) 1 ... [FL] Data Entry (1byte) 8	1-byte data entry(0-255) ... 1-byte data entry(0-255)
	147-154	2 bytes	I	W	0-FFFF	0	Previous	[FL] Data Entry (2bytes) 1 ... [FL] Data Entry (2bytes) 8	2-bytes Data entry (0-FFFF) ... 2-bytes Data entry (0-FFFF)
	155-159	1 bit	O	RT	0/1	0	Previous	[FL] Result of Function 1 (1bit) ... [FL] Result of Function 5 (1bit)	Result of FUNCTION 1 ... Result of FUNCTION 5
	160-164	1 byte	O	RT	0-255	0	Previous	[FL] Result of Function 1 (1byte) ... [FL] Result of Function 5 (1byte)	Result of FUNCTION 1 ... Result of FUNCTION 5
	165-169	2 bytes	O	RT	0-FFFF	0	Previous	[FL] Result of Function 1 (2bytes) ... [FL] Result of Function 5 (2 bytes)	Result of FUNCTION 1 ... Result of FUNCTION 5
					0°C-120°C	25°C	Previous	[FL] Result of Function 1 (2bytes) ... [FL] Result of Function 5 (2bytes)	Result of FUNCTION 1 ... Result of FUNCTION 5



BECOME USER!

<http://zennio.zendesk.com>

TECHNICAL SUPPORT