



KES 3xSingle-Phase (KNX Energy Saver)

Single-Phase System
Energy Saver

ZN110-KES

Application Program Version: [2.1]
User Manual Version: [2.1]_a

www.zennio.com

Contents

Document Updates.....	3
1 Introduction.....	4
1.1 KES, the KNX Energy Saver.....	4
1.2 Installation.....	5
1.2.1 Measuring Probe.....	7
2 Configuration.....	9
2.1 Active Power.....	9
2.2 Active Energy.....	9
2.3 Power Limits.....	10
3 ETS Parametrisation.....	12
3.1 Default Configuration.....	12
3.2 General Window.....	14
3.3 Channels.....	18
3.3.1 Active Energy.....	20
3.3.2 Active Power.....	23
3.3.3 Power Limit Monitoring.....	25
3.4 Logical Functions.....	29
ANNEX I: Time Management.....	30
ANNEX II: Further Examples.....	34
ANNEX III: Communication Objects.....	48

DOCUMENT UPDATES

Version	Changes	Pages(s)
[2.1]_a	Changes in the application program: <ul style="list-style-type: none"> • Higher precision in the measurement of the cost if tariff switches occur during the measuring period. • Improvement in the module of the Logical Functions to avoid sending unexpected values. 	
	Update of the Object Table.	48 - 49
	Revision of styles and minor clarification of certain texts.	-
[2.0]_a	Changes in the application program: <ul style="list-style-type: none"> • New algorithm for calculating the energy consumption. • Minor changes in texts. 	-

1 INTRODUCTION

1.1 KES, THE KNX ENERGY SAVER

The KNX Energy Saver from Zennio, **KES**, combines in only one device the following features:

- Support for **single-phase** and **three-phase** environments (two independent application programs):
 - **KES 1xThree-Phase**, destined to three-phase electric systems, where each of the measuring probes will monitor one of the phase lines.
 - **KES 3xSingle-Phase**, destined to single-phase electric systems, where the measuring probes can be used to monitor up to three independent electric conductions (channels).
- **3 independent input channels** for the connection of up to three measuring probes (model **ZN1AC-CST60**) to monitor three independent single-phase lines or one three-phase line (depending on the application program), which permits the following functions:
 - Measuring and reporting the **instant active power consumptions** (kW).
 - Calculating and reporting the **active energy consumptions** (kWh).
 - Calculating and reporting the cost of the energy consumption (with the possibility of setting up 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 the total energy consumption, as well as upon request (through a global request object).
- Report of the **actual and average values** of the consumed energy, the cost incurred and the corresponding CO₂ emissions (every day, every week, every

month or upon request).

- Synchronisation to an **external KNX clock**.
- Module with up to 5 multi-operation **logical functions**.

Note: *the measurement algorithm implemented in versions 2.0 and later of KES can correctly compute power consumptions in a variety of alternating currents, including non-sine signals and signals generated by dimmers. This makes KES able to work with signals of any shape and to measure power consumptions with all kinds of appliances and devices in the home and industrial environments.*



Figure 1. The KES Energy Saver.

1.2 INSTALLATION

KES connects to the KNX bus via the provided bus terminals.

The measuring probes need to be connected to the KES input channels through the screw terminal bundled within the device packaging.

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

Figure 2 shows the element scheme of KES.

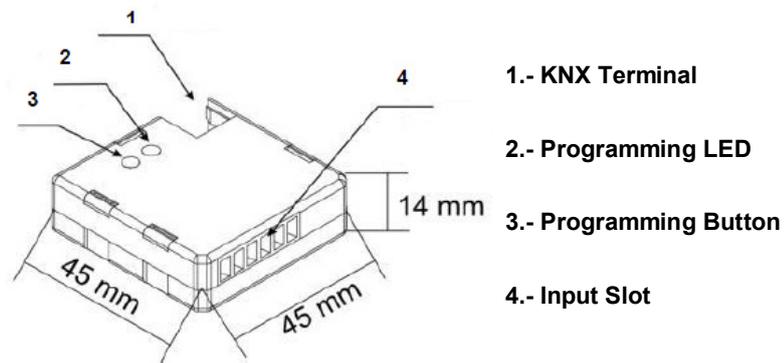


Figure 2 Element Scheme.

The functionality of the main elements is described below:

- **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, KES will enter the **Safe Mode**. The LED will then blink in red.
- **Input Terminal (4):** slot for the insertion of the screw input terminal (see figure below) and therefore for the connection of the different measuring probes provided with the device. A different probe will be required per electrical line being monitored. The two wires of each measuring probe will be inserted into the two connection points of the particular input channel to be used. For instance, if three electrical lines are going to be monitored, one probe will need to be connected to the slots 1 and 2 (referred to as “Channel A” in the Single-Phase application program), another one to the slots 3 and 4 (“Channel B”), and a third one to the slots 5 and 6 (“Channel C”). In the case of the Three-Phase version, each probe will be destined to monitor one of the three phase lines in the electrical system.

Note: *the insertion order of the two terminal wires from the measuring probe to the connection slots does not affect the functionality. I.e., for the first input channel it is possible to connect either one of the wires from the probe into point 1 and the other cable into point 2, or either the inverse.*

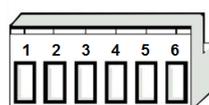


Figure 3. Screw Input Terminal.

1.2.1 MEASURING PROBE

The set-up process of the **ZN1AC-CST60** measuring probe (Figure 4) is described next.



Figure 4 Zennio ZN1AC-CST60 Measuring Probe.

The required steps are:

- **Open** the top clamp of the probe.
- Pass the **phase wire** from which the energy measures will be taken through the gap that has just been uncovered in the probe.

Important: *only one phase line should be inserted into the measuring probe, not the entire power cable with the neutral and other phases.*

- **Close** back the top clamp of the probe.
- Connect the **two wires** of the measuring probe into the two connection slots of the KES terminal block that correspond to the desired channel.

Please remember that KES only performs measurements of the power consumed by **alternating current** (AC) lines.

Note: *it is **very important** to keep in mind that neither ordinary power supplies (230 V) nor other external supplies should be connected to the KNX bus, nor directly to the input terminal of KES.*

For detailed information about the technical features of KES, as well as on security and installation procedures, please refer to the device **Datasheet**, bundled within the device packaging and also available at <http://www.zennio.com>.

Hereafter, this user manual will only refer to the **KES 3xSingle-Phase** application program, intended for single-phase systems. The specific user manual for the **KES 1xThree-Phase** application program, intended for three-phase systems, can be downloaded from the <http://www.zennio.com> website.

2 CONFIGURATION

2.1 ACTIVE POWER

KES 3xSingle-Phase calculates the value of the active power consumed by the device whose phase line passes through the ZN1AC-CST60 probe, connected itself to one of the input channels of KES. This active power is obtained from the **effective intensity** measured by the probe and according to the following relation:

$$P_{act} = V_{ef} \cdot I_{ef} \cdot PF$$

where V_{ef} is the effective voltage of the line and PF is the power factor (which is ideally assumed to be the equal and constant for the whole installation). Both values must be defined by parameter. The value of the effective intensity, I_{ef} , is obtained, as stated above, by placing the measuring probes around the electric line, and by plugging the outputs of the probe into one of the input channels of KES.

Each measuring probe, and consequently KES, is capable of measuring effective intensities in the range [300 mA - 60 A], which in a 230v single-phase environment with a power factor of 100%, would correspond to [69 W – 13,8 kW].

Once beyond the aforementioned 300 mA, the **resolution in the measurement of the instant power** is determined by the DPT of the 2-byte floating-point object used to transmit the value to the bus (0.01 kW).

In addition to the instant active power, KES can also monitor the highest power detected, so that **power peaks** can be calculated per hour, day, week and month, which may be interesting for monitoring historic values of power consumption.

2.2 ACTIVE ENERGY

Based on successive values of the instant power, KES can perform a 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 constant for the 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 for the production of such quantity of energy.

If both estimations and energy consumptions 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 power consumption in relation to certain limits, initially defined by parameter: an **upper limit** and a **lower limit** (linked to an overconsumption alarm and to a low-consumption indicator, respectively), and to a certain margin (**hysteresis** or **deadband**) regarding each of the two limits.

Figure 5 shows an example of the process. When the instant power is higher than the upper limit, KES sends an overconsumption alarm (if configured) which will not be deactivated until the instant power is back again under 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.

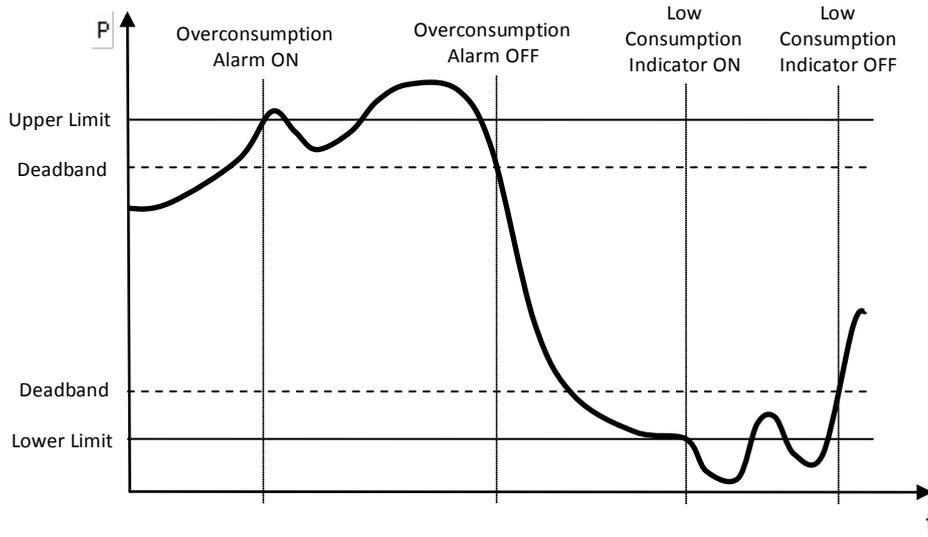


Figure 5 Power Limit Monitoring.

These concepts will be explained in detail in section 3.3.3 of this user manual.

3 ETS PARAMETRISATION

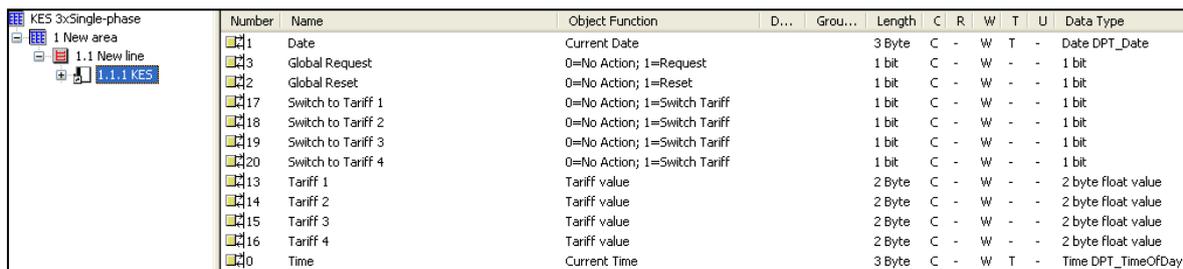
To begin with the parameterisation of KES it is necessary, once ETS is running, to import the product database (**KES 3xSingle-Phase** application program).

Next, the device must be added to the project where desired. And then, one right-click on the device will permit selecting "Edit parameters", in order to start the configuration.

The following sections describe in detail the different options available in ETS.

3.1 DEFAULT CONFIGURATION

When no particular parameterisation has been defined yet, the device and its objects will show as in Figure 6.



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 6 Default Topology.

Objects “**Time**” and “**Date**” are shown first in the list. Their purpose is to set the internal time and date of KES by means of an external KNX clock. Their functionality and everything related to the time management will be detailed in *ANNEX I: Time Management*.

“**Global Reset**” and “**Global Request**” are listed next. When the value “1” is received by the earlier, every counter and every calculation will be reset to their default values, so every object related to instant values, estimations, total values and average values that had been enabled by parameter will become zero, as well as every variable needed for calculation. In addition, communication objects related to tariffs and the upper and lower limits will return to their original values, that is, the one parameterised through the corresponding ETS fields. After that the measuring and calculating process will start over. The value “0” written to this object will have no effect.

Regarding “**Global Request**”, the value “1” through this object will cause an extra sending of any communication objects related to the power consumption –of every input channel that has been enabled– that, because of having been parameterised in such way, may be already being sent to the bus at certain times: instant values, accumulate values, average values, peak values, etc. In other words, a global request forces re-sending everything that KES, with no need of a specific external request, already sends periodically to the bus. Objects not being periodically sent will consequently not be sent, such as “Total Energy”, “Total Cost”, etc. In addition, when the global request takes place, objects for which a periodical sending is available but have not been parameterised in ETS will not be sent either.

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

3.2 GENERAL WINDOW

The General screen, shown in Figure 7, permits setting generic parameters related to the operation of the Energy Saver, as well as specifying which channels (A, B, C) have a ZN1AC-CST60 probe connected for measuring energy consumptions.

The screenshot shows the 'General Configuration' window with the following settings:

- AC Power Supply Voltage [V]: 230
- Power Factor [%]: 100
- Carbon Dioxide Ratio [x0.01 kgCO2/kWh]: 50
- Tariffs:
 - Tariff 1 Initial Value [x0.01 local currency/kWh]: 13
 - Tariff 2 Initial Value [x0.01 local currency/kWh]: 0
 - Tariff 3 Initial Value [x0.01 local currency/kWh]: 0
 - Tariff 4 Initial Value [x0.01 local currency/kWh]: 0
- Date and Time request on BUS voltage recovery:
 - Initial Delay [x1 sec]: 10
- Cyclical Sendings:
 - SECURITY Sending of last Daily values:
 - Cycle Time [x1 hour (0 = Disabled)]: 0
 - SECURITY Sending of last Weekly values:
 - Cycle Time [x1 day (0 = Disabled)]: 0
 - SECURITY Sending of last Monthly values:
 - Cycle Time [x1 day (0 = Disabled)]: 0
 - Sending of Daily, Weekly and Monthly Peak Power values:
 - Cycle Time [x1 hour (0 = Disabled)]: 0
- Reset after Request?: No
- Channels:
 - Channel A: No

Figure 7 Default Configuration Screen.

- **AC Power Supply Voltage [V]:** this field is to configure the effective voltage, in volts, of the electric network (0-450v). The default value is 230v.
- **Power Factor [%]:** defines, as a certain percentage, how the active power (the one actually consumed by the load) and the total supplied power are related. The default value is 100%.
- **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 the regular ratios in Europe. It is always possible to change it by entering values more accurate for the system being monitored or by directly setting the exact value that can usually be found on the electric bill provided by the power supplier.

- **Tariffs. Tariff “x”. Initial Value:** Initial value for tariff “x” (among up to four different tariffs) for the calculation of the costs (estimated, actual or average) of the energy consumptions. This value must be expressed in cents of the local currency.

KES permits defining up to four tariffs with different electric fees, and to switch from one to another in runtime (the active tariff by default is tariff 1). 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. These objects may be linked to an external timer so that, for example, depending on the time band in which KES takes consumption samples, one tariff or another is 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 “Switch to Tariff x” object can be left unlinked.

- **Date and Time Request on Bus Voltage Recovery:** defines how much time (1 to 255 seconds) KES must wait, whenever the bus voltage is recovered after a bus failure, 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 periodically sending certain objects (for every active channel). This includes both, a **Security Sending** (in other words, re-sending values that had already been calculated and reported to the bus but that can be sent again periodically) and a **Continuous Sending** of certain values, constantly updated. In order to better understand of how KES counts these periods, and the exact instant when the objects are sent, please refer to *ANNEX I: Time Management*.

- **Security Sending of Last Daily Values:** allows periodically re-sending the values related to the previous day, being possible to select cycle times of 1, 2, 3, 4, 6 or 12 hours (“0” disables this function). This way, all the daily values enabled under “Active Energy” (see section 3.3.1) 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), re-sending the last daily values will only be possible if at least one channel (A, B, C) is active and at least one daily value has been enabled under the “Active Energy” screen.*

- **Security Sending of Last Weekly Values:** permits periodically re-sending the values related to the previous week, being possible to select cycle times of 1 to 7 days (“0” disables this function). This way, all the weekly values enabled under “Active Energy” (see section 3.3.1) 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), re-sending the last weekly values will only be possible if at least one channel (A, B, C) is active and at least one weekly value has been enabled under the “Active Energy” screen.*

- **Security Sending of Last Monthly Values:** permits periodically re-sending values related to the previous month, being possible to select cycle times of 1 to 30 days (“0” disables this function). This way, all the monthly values enabled under “Active Energy” (see section 3.3.1) 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), re-sending the last monthly values will only be possible if at least one channel (A, B, C) is active and at least one monthly value has been enabled under the “Active Energy” screen.*

- **Sending of Daily, Weekly and Monthly Peak Power Values:** allows periodically sending the instant (updated) values of the daily, weekly and monthly peak power, being possible to select a cycle time of 1, 2, 3, 4, 6

or 12 hours (“0” disables this function). This way, the peak power values enabled under the “Active Power” screen (see section 3.3.1) will not only be sent when corresponding (at the end of the day, at the end of week, or at 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, 4, 6 or 12 hours), sending the peak power values will not be possible unless 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 this parameter is enabled (“Yes”), the value of the communication objects and variables related to the total consumption of a particular channel (A, B, C) will be reset to zero every time KES responds to a request of the total consumption values of that channel since the last reset. This local request is performed by sending the value “1” to the “[Cx] Request” object (see section 3.3).
- **Channels A, B and C:** permits selecting what channels will be used for measuring. Once enabled (“yes”), the menu on the left will include a tab for the configuration of each active channel (see section 3.3).
- **Logical functions:** after enabling this parameter, the menu on the left will include a new tab for the configuration of the logical functions. See section 3.4 for further information.

3.3 CHANNELS

Channel screens will show for the first time as follows:

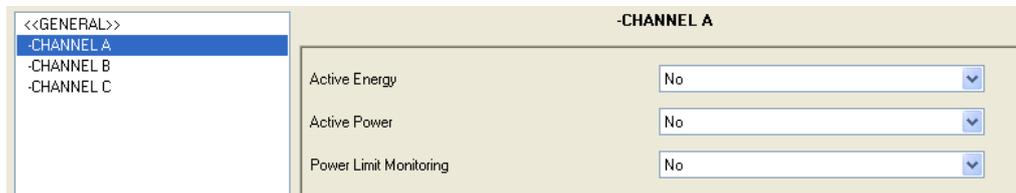


Figure 8 Channel.

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 to only a certain channel).

- When the Energy Saver receives the value “1” through the “[Cx] Reset” object, every communication object and variables related to estimates and average / total values of channel “x” will be reset to zero. Receiving the value “0” will have no effect.
- When the Energy Saver receives the value “1” through the “[Cx] Request” object, it will report the accumulate value –since the last reset– of the energy consumption in channel “x”, as well as the incurred cost and the carbon dioxide emissions. These values are reported through the “[Cx] Total Energy”, “[Cx] Total Cost” and “[Cx] Total CO2 Emissions” communication objects respectively, which will automatically show up when the corresponding option gets enabled in ETS. In addition, if “Reset after request” has been enabled under the General screen, whenever a “1” is received through the “[Cx] Request” object, a reset process similar to that described above will also take place.

Note: *the three Total objects are not intended to be read independently; their values will only be updated and sent to the bus all together, as a response to the “[Cx] Request” object.*

- “[Cx] Disable” allows enabling or disabling the power measurement in channel “x” (according to the KNX standard, “0” corresponds to disabling the measurement and “1” corresponds to enabling it). When this measurement has been disabled, the cyclical sending of the instant power is interrupted (in case it had been previously parameterised), and the consumption is considered zero from that moment. Note that deactivating this measurement is not equivalent to simply not powering the device (which would make it not aware of time passing), but to measuring a null power consumption while time is still passing. In order to clarify this point, reading *ANNEX II: Further Examples* is encouraged (particularly Example II).

On the other hand, when the power measuring process is disabled, some more sending is interrupted, according to the table shown below.

	“Disable” = 0	“Disable” = 1
Cyclic sending of instant power		
Sending of peak power		
Cyclic re-sending of peak power		
Sending of effective consumptions		
Cyclic re-sending of effective consumptions		
Cyclic sending of estimated consumptions		
Sending of average consumptions		
Time passed counts for average values?	(1)	
Real consumption counts for calculations?		

Table 1 Sending vs. Activation/Deactivation of Power Measures

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.

¹ See Example II of ANNEX II: Further examples.

Note: the screen captures shown will refer to the configuration of Channel B. Similar parameterisation 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 are shown grouped into sections, as explained below.

- Estimated Values.** Permits the configuration of parameters related to the estimation of the active energy consumption, expressed in kWh, as well as the estimated cost, in terms of the local currency. This estimation of the active energy consumed is made by assuming that the instant power consumption at the moment of the estimation remains constant during the 24 hours of the current day.

Figure 9. Active energy. Estimations.

Related to estimations, two communication objects come up: “[Cx] Estimated Energy Consumption” and “[Cx] Estimated Cost”, being possible to configure the following:

- Minimum Time Between Sendings:** allows setting the minimum time, in seconds, that should pass between the sending of every two consecutive values of the corresponding object (“Estimated Energy Consumption” and/or “Estimated Cost”), no matter if they correspond to a cyclic sending or to a sending due to a change of the value. Valid

times are [5-255] seconds. This minimum time between the two will only be taken into account if a value other than “0” is parameterised for the 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: *if the value specified here (x) is lower than the minimum time between sendings (y), the latter will prevail, resulting in a sending taking place every “y” seconds. If “0” is parameterised 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 (either 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, this field can be left with value “0”.

Note: *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 “Global Request” communication object.*

- **Hourly Values.** From this section it is possible to enable the sending of the 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 be shown 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, the related 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 the 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

Note: *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, the incurred cost and the 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

Note: 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, the incurred cost and the required CO₂ emissions can be enabled from here, as well as of 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

Note: 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.

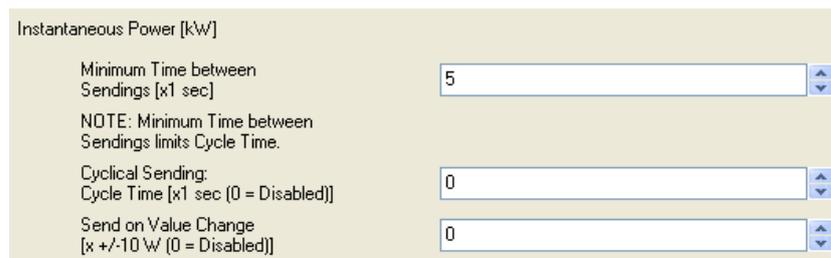
3.3.2 ACTIVE POWER

Active 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
Peak Power Values	

Figure 10 Active Power.

From this screen it is possible to configure several parameters related to the consumption of active power (section 2.1). These are shown grouped into different sections, as explained below:

- **Instantaneous Power.** This section is for configuring the parameters that affect the sending of the “[C] Instantaneous Power” communication object, whose value is expressed in kW with two decimal figures.



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:** permits setting the minimum time, in seconds, which should pass between the sending of two consecutive values of the “Instantaneous Power” object, no matter if they correspond to a cyclical sending or to a sending due to a change of the value. Valid times are [5-255] seconds. This minimum time between the two will only be taken into account if a value other than “0” is parameterised for fields “Cycle Time” or “Send on Value Change”.
- **Cycle Time:** allows setting the period for sending the “Instantaneous Power” object, in seconds (0 – 255). In other words, every how much time these values will be sent to the KNX bus.

Note: if the value specified here (x) is lower than the minimum time between sendings (y), the latter will prevail, resulting in a sending taking place every “y” seconds. If “0” is parameterised 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 the difference between two consecutive measurements of the instant power is found to be greater than that value, a new sending of the corresponding communication object (“Instantaneous Power”) will happen. If there is no need of this feature, this field can be set to 0.

Note: if a value other than “0” is set for “Cycle Time” and “Send on Value Change”, the “Instantaneous Power” object will also be sent to the KNX bus whenever the value “1” is received through the “Global Request” communication object.

- 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 “Instantaneous Power”, making it possible to inform the bus about how the hourly peak value changes throughout the hour. Regarding the 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, week or month), however in this case this must be configured under the General screen, with a unique parameterisable 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 monitoring the power consumption in relation to upper and lower limits (section 2.3). Two different

sections can be found, one to configure the parameters of the upper limit and one to configure the parameters of the lower one.

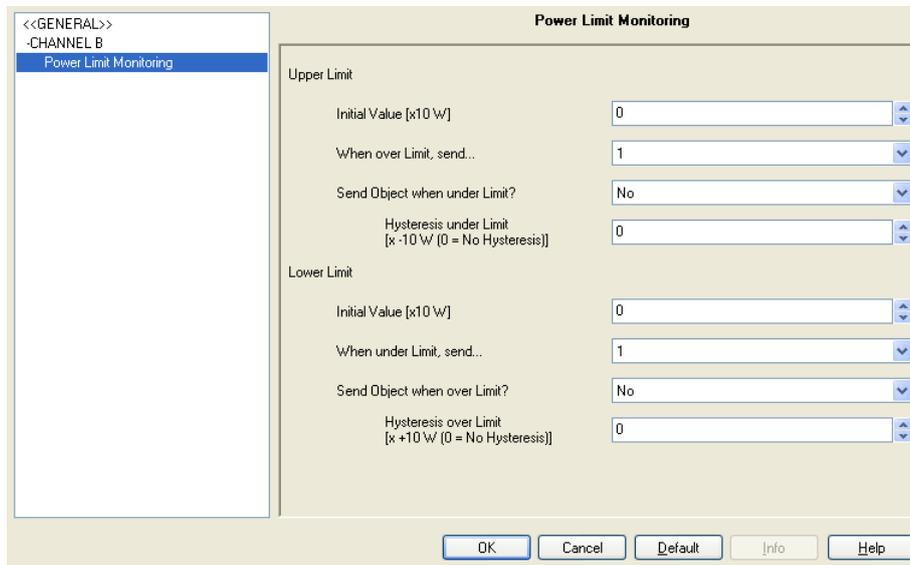


Figure 11 Power Limit Monitoring.

For the upper limit:

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

- Initial Value:** allows setting (in tens of watts) the value that, at least in the beginning, the Energy Saver will assume as the upper limit while measuring the power. If this is set to “0”, KES will not monitor whether the power consumption exceeds or not a particular upper limit. The value of the upper limit, however, can be modified in runtime through the “[Cx] Upper Limit” communication object. In the case of setting a value other than “0” in this field, it will be as well necessary to configure the parameters described below, which define how the upper limit monitoring behaves.
- When Over Limit, Send:** allows setting the value (“0” or “1”) that will be written to the “[Cx] Over-Consumption Alarm” object when the instant power becomes greater than the value defined as the upper limit, making the user aware of this over-consumption situation in the system.

- **Send Object when Under Limit?:** allows specifying whether the “[Cx] Over-Consumption Alarm” object should or should not send 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.
- **Hysteresis Under Limit:** this parameter can be used to avoid sending 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 but then exceeds it again. By configuring a certain margin with a value other than “0” (in tens of watts), 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 5 (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
Hysteresis over Limit [x +10 W (0 = No Hysteresis)]	0

- **Initial Value:** allows setting (in tens of watts) the value that, at least in the beginning, the Energy Saver will assume as the lower limit while measuring power. If this is set to “0”, KES will not monitor whether the power consumption becomes or not lower than a particular lower limit. The value of the lower limit, however, can be modified in runtime through the “[Cx] Lower Limit” communication object. In the case of parameterising a value other than “0” for this field, it will be as well necessary to configure the parameters described below, which define how the lower limit monitoring behaves.
- **When Under Limit, Send:** allows setting the value (“0” or “1”) that will be written to the “[Cx] Low Consumption Indicator” object when the instant power becomes lower than the value defined as 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 “[Cx] Low Consumption Indicator” object should send or not the inverted value (i.e., “0” if the previous parameter was set to “1”) once the instant power is 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..
- **Hysteresis Over Limit:** this parameter can be used to avoid sending consecutive low-consumption indicators every time the instant power –after having become lower than the lower limit and therefore having activated the indicator– is no longer lower than the limit but then becomes lower again. By configuring a certain margin with a value other than “0” (in tens of watts), every time the low-consumption indicator is triggered, it will be necessary that the measured power reaches *lower limit plus margin* before further low consumptions trigger the indicator again. See Figure 5.

3.4 LOGICAL FUNCTIONS

This module is provided to perform arithmetical operations or in binary logic 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 purpose.

Up to five different (and independent of each other) **logical functions** can be enabled, carrying out a maximum of **four operations** each. To use any of them, it is necessary to configure the following ETS window, which appears after selecting “Yes” in the Logical Functions box of the General window.

Function	Status
- FUNCTION 1	Enabled
- FUNCTION 2	Disabled
- FUNCTION 3	Disabled
- FUNCTION 4	Disabled
- FUNCTION 5	Disabled

Data Entry Object	Value
- 1bit	0
- 1byte	0
- 2bytes	0

Figure 12 Logical functions.

To obtain detailed information about using and parameterising the logical functions, please refer to the "**Logical Functions X5**" specific document, available from Zennio at <http://www.zennio.com>.

ANNEX I: TIME MANAGEMENT

As shown in section 3.1, the objects “Date” and “Time” are displayed by default in ETS during the device configuration. The purpose of these two objects is to set KES to the date and time of an external KNX clock. However, KES features its own built-in clock, so these two objects are typically only necessary for occasional synchronisation of 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 normal circumstances it is not necessary that the KNX clock sends these objects continuously.

Consequently, every time KES is programmed or recovers after a bus voltage failure, two read requests are sent to the bus through these objects, with the intention of updating the internal clock to the time and date that will be received as a response from the KNX external clock. If this request receives no response after five seconds, KES will retry 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 just been programmed, that the current date is January 1st 2000 and the current time is 0:00 am–.

Regarding the cyclical sending of objects –both the one enabled from the General screen and the one enabled from that of the channel itself–, 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, 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 period counter is reset at the end of the time unit immediately greater** than the one used for defining the sending periods. The following example illustrates this:

Example III:

Exactly at midnight, the daily values corresponding to the previous day will always be calculated and sent (in particular, those daily values that have been enabled in ETS). If a security sending of the last daily values has been parameterised for every four hours, the hour counter will be reset at midnight. That is 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 to Monday) the weekly values corresponding to the previous week will always be calculated and sent (in particular, those weekly values that have been enabled in ETS). If a security sending of the last weekly values has been parameterised for every four days, the day counter will be reset during such 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 sending takes place on the same days every week (in the example, when switching from Thursday to Friday, apart from the ordinary sending at the end of the Sunday), instead of spreading a certain delay from week to week.

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

Example V:

A security sending is parameterised for every six hours. A parameter download takes place at 9:25h am, after which the internal clock is set to the current time. When it is 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 the current time (otherwise, it would assume that it is still 5:15h pm). Since the voltage failure has caused a reset in the timing of the periodical sending, the next security sending will happen at 11:00h pm, even if the previous one had taken place at 3:00h pm.

Finally, and for 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 during tests, it is important to note that **the timers in KES do not really count the number of hours going by, but the number of forward hour changes**. Of course, all considerations that have already made will also apply here, as shown in the following examples:

Example VI:

A periodic sending has been parameterised for every three hours and a parameter download takes place at 9:25h am, after which the internal clock of KES is set to the current time. Five minutes later, 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 throw 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 stated 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 parameterised for every three hours. The device has been turned on and the internal clock has been correctly set. As soon as it is 3:00h am the periodic sending takes place. However, assuming that Today is the day when the clock jumps backward an hour for daylight saving (Summer to Winter time switch), after such sending the external KNX clock will send an order to the internal clock to jump back to 2:00h am. This backtracking does not affect the timing of the periodic sending, 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 happened). 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 there will not be a sending at 5:00h am anymore, but at 6:00h am.

ANNEX II: FURTHER EXAMPLES

With the aim of making clear how KES works and of presenting the multiple parameterisable communication objects in their context, a few more examples are offered next. They are intended to cover most of the doubts that may arise 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 them, KES may report incorrect or null values through the objects sent to the bus (such as “Last hour Energy Consumption”).*

Example I: *total consumption.*

The simplest application of KES consists in simply calculating total consumptions. In that case, it would be enough to parameterise something like the following:

- **General screen:** *only one tariff, with a cost of 15 cents per kWh, and a CO₂ ratio of 50. Voltage, the one corresponding to the system. Power factor, 100. Enable the corresponding channel. Cyclical sending 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, KES will perform the following functions:

- 1) *Sending every midnight the “Daily Energy Consumption” and “Daily Cost” objects, expressed in kWh and monetary units, respectively.*
- 2) *Sending, on the 1st day of every month at 0:00h am, the “Monthly Energy Consumption” and “Monthly Cost” objects, expressed in kWh and monetary units respectively.*

For simplicity, the probe can be supposed to be measuring the consumption of a 100-watt bulb throughout a certain month with 22 workable days –during which the bulb stays on between 8:00h in the morning and 6:00h in the afternoon– and 8 non-workable days –during which the bulb remains off–. In such case:

- 1) Every midnight, the “Daily Energy Consumption” object is sent to the bus with value “0” when the previous day has been non-workable, or $(8h * 0.100 kW) = 0.8 kWh$ if workable, while the “Daily cost” object will have a value of 0 after a non-workable day and of $(0.8 kWh * 0.15 m.u.) = 0.12$ monetary units after any other day.
- 2) After every last day of a month, at midnight, the “Monthly Energy Consumption” object sent to the bus will have a value of $(0 kWh * 8 days) + (0.8 kWh * 22 days) = 17.6 kWh$, while the “Monthly Cost” will be $(17.6 kWh * 0.15) = 2.64$ monetary units.

This way, after every month KES will automatically estimate in what grade the bulb contributed to the consumption and to the electric expenses.

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

Last, there is always the possibility of making global or local requests (through the “Global Request” and “[Cx] Request”) objects at the end of the month so that KES reports the value of “the Total cost” and of the “Total Energy”. This procedure is somehow similar to reading the ordinary counter installed by the power supplier: it would be enough to calculate the difference between the current value of “Total Energy” and that from the 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 is why the 0.8 kWh mentioned in the above example may show as 1 kWh, although the 0.8 kWh are actually considered as such for calculation.

Example II: channel disabled.

Four different cases are presented next in relation to how KES works when the channel where the measurement is being made remains 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

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

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

● **CASE 2**



Similar situation, but one of the days the channel is left disabled for 12 hours (no matter which). 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 remains at 3 kW, then 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. In such case, that day:

[Cx] Daily Energy Consumption = 0 kWh
 [Cx] Average Daily Energy Consumption = 0 kWh

If the channel is enabled again for the rest of the week and the power consumption remains at 3 kW, that particular day will not be taken into account when the week ends:

[Cx] Weekly Energy Consumption = 432 kWh
 [Cx] Average Weekly Energy Consumption = 432 / 6 = 72 kWh ≈ 67 kWh

● CASE 4



In the case of having no consumption during a whole day (note that a null consumption is a very remote situation) although the channel stayed enabled, KES will do take into account that day for the weekly calculation. For example, assume that there was a daily consumption of 72 kWh during the rest of the week:

[Cx] Daily Energy Consumption = 0 kWh

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

[Cx] Weekly Energy Consumption = 432 kWh

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

● CASE 5



Another particular case is that of having, during one day, a total consumption of zero (note that such situation is quite improbable) and the channel enabled for only 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 to be active.

If the channel stays disabled that day only 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 stays disabled that day only between 7:00h pm and 12:00h 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

(Assume, for the remaining six days in the week, channel enabled and daily consumption of 72 kWh).

Example III: *channel disabled.*

While measuring the electric consumption of a certain device that remains always turned on and that consumes 200 W constantly, the corresponding channel of KES is left enabled between 0:00h am and 11:00h am on Monday. Exactly at 11:00h am, the channel is deactivated through the “[Cx] Disable” object. When it gets to 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.

Suppose the device whose consumption is being measured now stays turned on only on Tuesday, Thursday and Saturday and that the channel is 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, including those in which there was a consumption of exactly zero.

Example IV: *channel disabled.*

Suppose the energy consumption of an office is going to be monitored. 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 for 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 by 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 to be disabled at the end of the day (midnight) will be discarded.

Example V: tariffs.

Let the following tariff parameterisation be downloaded 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 gets switched on at midnight.

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

Example VI: *power factor.*

Assume a system where the power factor is 90%, i.e., where 10% of the effective energy (the power 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 parameterised, 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 the effective energy that is really transmitted by the monitored line is a little greater.
- If a power factor of 100% is parameterised, 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 parameterisation with a power factor of 80% will cause an excessive correction made by KES, which will send the bus smaller values than the effective ones but also than the active ones.

Example VII: *power limits.*

Assume a Power Limit parameterisation as follows:

- *Upper limit:*

*Initial Value = 200 (tens of watts)
When Over Limit, Send = 1
Send Object when Under Limit = Yes
Hysteresis Under Limit = 20 (tens of watts)*

- *Lower limit:*

*Initial Value = 100 (tens of watts)
When Under Limit, Send = 1
Send Object when Over Limit? = Yes
Hysteresis Over Limit = 20 (tens of watts)*

Under this parameterisation, the following can be observed:

- 1) As soon as the instant power is over 2 kW, a “1” will be sent through “[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 “[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 one “1” through “[Cx] Low Consumption Indicator”.
- 5) A new value of 1.05 kW will not have an effect, nor will 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 “[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 “[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 practical sense.

Example VIII: global test.

This practical example aims to simulate in only a few minutes how KES would work in a longer time lapse (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 parameterisation is assumed:

- **General screen:** all cyclic and security sendings, enabled and with the shortest cycle time (i.e., the most frequent): 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. The voltage corresponding to the system, and a power factor of 100.
- **Active Energy screen:** values 5, 30 and 0 for “Estimated Energy Consumption”. Values 5, 30 and 0 for “Estimated Cost”. Any other parameters, enabled.
- **Active Power screen:** values 5, 30 and 0 for section “Instantaneous Power”. Values 10, 30 and 0 for section “Hourly Peak Power”. Any other parameters, enabled.

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

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

30 sec. after the time set, the Instantaneous power will be sent with a value of 2 kW.

Estimated Energy Consumption ≈ 48 kWh.

This is calculated by multiplying 2 KW by 24 hours, which indicates the estimated daily consumption assuming the current instant power as constant.

Estimated Cost ≈ 2400 monetary units.

This is obtained by multiplying 48 kWh by the parameterised 50 monetary units.

Hourly Peak Power ≈ 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 at 10:00h am.

After the time change, Last Hour Energy Consumption will be sent with 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 while during the remaining five minutes 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 ≈ 50 monetary units.

50 is the cost corresponding to the one 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. This is why the total consumption for Monday is estimated to be $1 \text{ kWh} + 2 \text{ kWh} = 3 \text{ kWh}$.

Daily Cost ≈ 150 monetary units.

$3 \text{ kWh} * 50 \text{ m.u.} = 150 \text{ m.u.}$

Daily CO₂ Emissions $\approx 1.5 \text{ kg}$

Obtained as $(50 \text{ kg} * 0.01) * 3 \text{ kWh}$.

Daily Peak Power $\approx 3 \text{ kW}$.

This is the highest instant power value detected during the previous day.

Average Daily Energy Consumption = 2 kWh.

This value represents the energy amount that, as an average, was consumed per hour during the previous 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 ≈ 0.75 .

This value is the result of $(50 \text{ kg} * 0.01) * 3 \text{ kWh} / 2 \text{ hours}$.

Average Daily Cost ≈ 75 monetary units.

This object indicates the cost that, as an average, will be paid per hour 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, the last week was only 2 day long (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 kg.

This value is the result of calculating $(50 \text{ kg} * 0.01) * 2 \text{ kWh} / 1 \text{ hour}$.

Average daily cost \approx 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 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 \approx 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 kWh in total / 1 day = 2 kWh per day.

Average Weekly CO₂ Emissions ≈ 1 kg per day.

$(50 \text{ kg} * 0.01) * 2 \text{ kWh} / 1 \text{ day} = 1.$

Average Weekly Cost ≈ 100 monetary units per day.

2 kWh * 50 monetary units / 1 day = 100 m.u.

Weekly Peak Power ≈ 2 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 ≈ 450 monetary units.

In other words, $(50 \text{ kg} * 0.01) * 9 \text{ kWh}.$

Monthly Peak Power ≈ 3 kWh.

This value is due to the temporary connection of the 3000 W load.

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 ≈ 1.125 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 then divided by only one day.

Average Monthly Cost $\approx 100 \text{ m.u. per day}$.

Those 2 kWh consumed throughout the month have an associated cost of 50 u.m each. Then, this is divided by only one day.

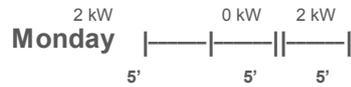
Average Monthly CO₂ Emissions $\approx 1 \text{ kg. per day}$.

$(50 * 0.01) * 2 \text{ kWh} / 1 \text{ day} = 1$.

Simulation Diagram (Example VIII)

Month 1

WEEK 1



Daily Consumption = 1 kWh + 2 kWh = 3 kWh



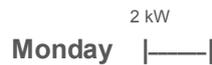
Daily Consumption = 2 kWh

WEEK 2



Daily Consumption = 2 kWh

WEEK 3



Daily 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



Daily consumption = 2 kWh

Monthly Energy Consumption = 2 kWh.

Average Monthly Energy Consumption = 2 kWh / 1 day = 2 kWh per day.

ANNEX III: 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.
- **“1st boot”** shows the cases where an object is assigned a certain value by the application program after a device download or a full reset. In case the value of such assignment can be parameterised, ✓ is shown in column **“P”**. Objects showing a hyphen (-) are not assigned a particular value and therefore can be assumed to be initialised with the value “0” or with the corresponding updated value in case they depend on an external element (sensors, etc.). Moreover, if the object is sent (or is there an option to send it) to the bus (write or read requests) after a download or a device reset from ETS, the marks **(W)** or **(R)** will be shown, respectively for transmissions or read requests.
- **“Reboot”** shows the cases where an object is assigned a certain value by the application program after a bus power failure. In case the value of such assignment can be parameterised, ✓ is shown in column **“P”**. Objects showing a hyphen (-) are not assigned a particular value and therefore can be assumed to maintain their previous value after the failure or with the corresponding updated value in case they depend on external elements (sensors, etc.). Moreover, if the object is sent (or is there an option to send it) to the bus (write or read requests) after a bus failure, the marks **(W)** or **(R)** will be shown, respectively for transmissions or read requests.

Number	Size	I/O	Flags	Data Type (DPT)	Functional Range	1st time	P	Reboot	P	Name	Function
0	3 Bytes	I	C T - W U	DPT_TimeOfDay	00:00:00 - 23:59:59	- (R)		- (R)		Time	Current Time
1	3 Bytes	I	C T - W U	DPT_Date	01/01/1990 - 31/12/2089	- (R)		- (R)		Date	Current Date
2	1 Bit	I	C - - W -	DPT_Reset	0/1	-		-		Global Reset	0=No Action; 1=Reset
3	1 Bit	I	C - - W -	DPT_Trigger	0/1	-		-		Global Request	0=No Action; 1=Request
4-6	1 Bit	I	C - - W -	DPT_Reset	0/1	-		-		[CX] Reset	0=No Action; 1=Reset
7-9	1 Bit	I	C - - W -	DPT_Trigger	0/1	-		-		[CX] Request	0=No Action; 1=Request
10-12	1 Bit	I	C - - W -	DPT_Enable	0/1	-		-		[CX] Disable	0=Disable; 1=Enable
13-16	2 Bytes	I	C - - W -	Non-standardized datapoint type	- 670760.00 - 670760.00	-	✓	-		Tariff X	Tariff value
17-20	1 Bit	I	C - - W -	DPT_Trigger	0/1	-		-		Switch to Tariff X	0=No Action; 1=Switch Tariff
21-23	4 Bytes	O	C T R - -	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Estimated Energy Consumption	Estimated Energy Consumption
24-26	2 Bytes	O	C T R - -	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Estimated Cost	Estimated Cost
27-29	4 Bytes	O	C T R - -	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Last Hour Energy Consumption	Last Hour Energy Consumption
30-32	2 Bytes	O	C T R - -	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Last Hour Cost	Last Hour Cost
33-35	4 Bytes	O	C T R - -	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Daily Energy Consumption	Daily Energy Consumption
36-38	2 Bytes	O	C T R - -	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Daily Cost	Daily Cost
39-41	2 Bytes	O	C T R - -	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Daily CO2 Emissions	Daily CO2 Emissions

Number	Size	I/O	Flags	Data Type (DPT)	Functional Range	1st time	P	Reboot	P	Name	Function
42-44	4 Bytes	O	CTR--	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Average Daily Energy Consumption	Avg. Daily Energy Consumption
45-47	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Average Daily Cost	Average Daily Cost
48-50	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Average Daily CO2 Emissions	Average Daily CO2 Emissions
51-53	4 Bytes	O	CTR--	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Weekly Energy Consumption	Weekly Energy Consumption
54-56	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Weekly Cost	Weekly Cost
57-59	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Weekly CO2 Emissions	Weekly CO2 Emissions
60-62	4 Bytes	O	CTR--	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Average Weekly Energy Consumption	Average Weekly Energy Cons.
63-65	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Average Weekly Cost	Average Weekly Cost
66-68	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Average Weekly CO2 Emissions	Average Weekly CO2 Emissions
69-71	4 Bytes	O	CTR--	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Monthly Energy Consumption	Monthly Energy Consumption
72-74	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Monthly Cost	Monthly Cost
75-77	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Monthly CO2 Emissions	Monthly CO2 Emissions
78-80	4 Bytes	O	CTR--	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Average Monthly Energy Consumption	Average Monthly Energy Cons.
81-83	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Average Monthly Cost	Average Monthly Cost
84-86	2 Bytes	O	CTR--	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Average Monthly CO2 Emissions	Average Monthly CO2 Emissions
87-89	4 Bytes		CT---	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[CX] Total Energy	Total Energy Consumption
90-92	2 Bytes		CT---	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Total Cost	Total Cost
93-95	2 Bytes		CT---	Non-standardized datapoint type	- 670760.00 - 670760.00	-		-		[CX] Total CO2 Emissions	Total CO2 Emissions
96-98	2 Bytes	O	CTR--	DPT_Power	- 670760.00 - 670760.00	-		-		[CX] Instantaneous Power	Instantaneous Power
99-101	2 Bytes	O	CTR--	DPT_Power	- 670760.00 - 670760.00	-		-		[CX] Hourly Peak Power	Hourly Peak Power
102-104	2 Bytes	O	CTR--	DPT_Power	- 670760.00 - 670760.00	-		-		[CX] Daily Peak Power	Daily Peak Power
105-107	2 Bytes	O	CTR--	DPT_Power	- 670760.00 - 670760.00	-		-		[CX] Weekly Peak Power	Weekly Peak Power
108-110	2 Bytes	O	CTR--	DPT_Power	- 670760.00 - 670760.00	-		-		[CX] Monthly Peak Power	Monthly Peak Power
111-113	2 Bytes	I	C--W-	DPT_Power	- 670760.00 - 670760.00	-	✓	-		[CX] Upper Limit	Upper Limit value
114-116	1 Bit	O	CTR--	DPT_Trigger	0/1	-		-		[CX] Over-Consumption Alarm	Alarm -> Sending of "0" or "1"
117-119	2 Bytes	I	C--W-	DPT_Power	- 670760.00 - 670760.00	-	✓	-		[CX] Lower Limit	Lower Limit value
120-122	1 Bit	O	CTR--	DPT_Trigger	0/1	-		-		[CX] Low Consumption Indicator	Ind. -> Sending of "0" or "1"
123-138	1 Bit	I	C--W-	DPT_Switch	0/1	-		-		[LF] (1 bit) Data Entry X	Binary Data Entry (0/1)
139-146	1 Byte	I	C--W-	DPT_Value_1_Ucount	0 - 255	-		-		[LF] (1 byte) Data Entry X	1 byte Data Entry (0-255)
147-154	2 Bytes	O	CTR--	DPT_Value_2_Ucount	0 - 65535	-		-		[LF] Function X RESULT (2 bytes)	FUNCTION X Result
	2 Bytes	O	CTR--	DPT_Value_Temp	0-120.00	-		-		[LF] Function X RESULT (2 bytes)	FUNCTION X Result
155-159	1 Bit	O	CTR--	DPT_Switch	0/1	-		-		[LF] Function X RESULT (1 bit)	FUNCTION X Result
160-164	1 Byte	O	CTR--	DPT_Value_1_Ucount	0 - 255	-		-		[LF] Function X RESULT (1 byte)	FUNCTION X Result
165-169	2 Bytes	O	CTR--	DPT_Value_2_Ucount	0 - 65535	-		-		[LF] Function X RESULT (2 bytes)	FUNCTION X Result
	2 Bytes	O	CTR--	DPT_Value_Temp	0-120.00	-		-		[LF] Function X RESULT (2 bytes)	FUNCTION X Result

Join and send us your inquiries
about Zennio devices:

<http://zennioenglish.zendesk.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