



# KCI 4 S0

## KNX Consumption Interface

**KNX-S0 Interface with 4 Inputs**

**ZRX-KCI4S0**

Application program version: [1.0]  
User manual edition: [1.0]\_a

[www.zennio.com](http://www.zennio.com)

## Contents

1	Introduction .....	3
1.1	KCI 4 S0 .....	3
1.2	Installation .....	4
2	Configuration .....	6
2.1	General.....	6
2.2	Input S0x .....	8
2.2.1	Configuration .....	8
2.3	Initialisation and Power Loss.....	16
	ANNEX I: Power Calculation .....	18
	ANNEX II. Communication Objects .....	21

# 1 INTRODUCTION

---

## 1.1 KCI 4 S0

---

KCI 4 S0 from Zennio is an S0 to KNX interface that makes it possible to integrate S0-pulse generators into the KNX installation. Specifically, **up to four S0-pulse generators** can be connected independently to the inputs of KCI 4 S0, which will interpret these S0-pulses as energy, water or gas consumption measurements, depending on the external S0-compliant meter that has been connected.

Therefore, KCI 4 S0 brings the possibility of notifying other KNX devices about the **consumption levels** measured by external S0-pulse generators. In practice, KCI may still manage to monitor a flow or caudal of any other type (not only electric power, water and gas) as long as it is measured by a pulse generator (and even non-S0 pulses may be recognised). However this is not guaranteed – previous testing is encouraged.

The most outstanding features of KCI 4 S0 are:

- 4x inputs for monitoring **up to four S0-pulse generators**.
- Customisation of the **pulse frequency**, i.e., how many pulses are expected per unit consumption.
- Acquisition of the electric energy, water or gas **consumption**, with the possibility of sending it to the KNX bus both periodically or on value changes.
- Calculation of the **electric power** and of the **water/gas flow rate**, with the possibility of sending them to the KNX bus both periodically or on value changes.
- Possibility of customising the **initial consumption value**, if not zero.
- **Auxiliary battery pack** to ensure the pulse count even in the absence of bus power.
- LEDs and communication objects to indicate **low or empty battery** situations.

## 1.2 INSTALLATION

KCI 4 S0 connects to the KNX bus through the on-board KNX connector. Once the device is provided with power from the KNX bus, both the individual address and the associated application program can be downloaded.

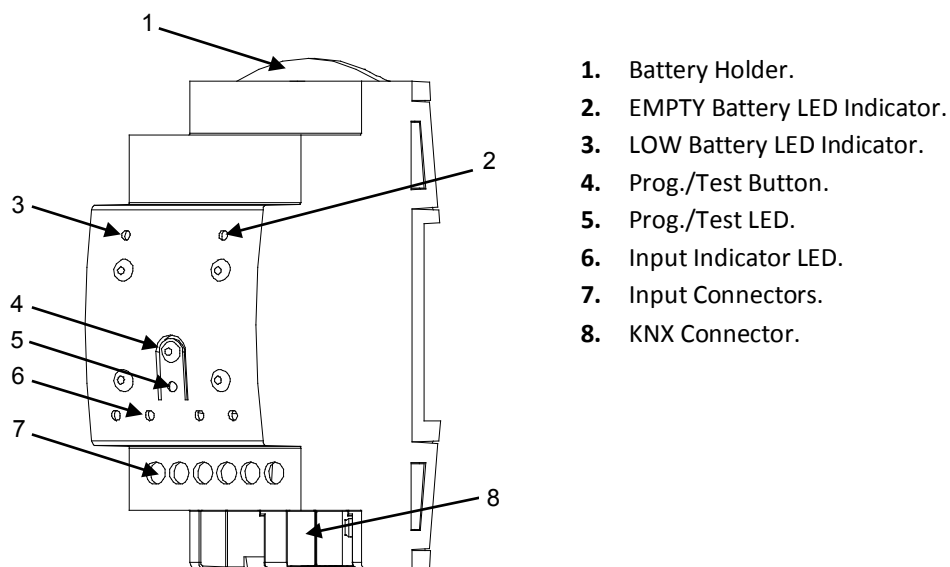


Figure 1. KCI 4 S0 - Element Diagram.

The main elements of the device are described next.

- **Test/Prog. Pushbutton (4):** a short press on this button sets the device into the programming mode, making the associated LED (5) light in red.

**Note:** if this button is held while plugging the device into the KNX bus, the device will enter into **safe mode**. In such case, the LED will blink in red.

- **Input Connectors (7):** slots for the connection of the input lines from the pulse generators. Whenever a pulse is received through one of the inputs, the associated LED (6) will flash briefly.
- **Battery Holder (1):** host for the two CR2032 batteries (2 x 3V) that will provide power in the absence of the KNX bus.
- **Low-Battery Indicator (3):** LED indicator that will blink in red if the battery level is weak. It is advisable to change the batteries when this happens.

- **Empty-Battery Indicator (2):** LED indicator that will blink in red if the battery level is critically low. It is required to change the batteries when this happens.

**Notes regarding the low/empty battery indicators:**

- *The two LED indicators are mutually exclusive. Once the empty-battery indicator turns on, the low-battery indicator will turn off.*
- *In the absence of KNX bus power, the device will keep trying to count pulses even in the empty-battery state. However, once under the low-battery level, an accurate behaviour cannot be guaranteed.*
- *To extend the battery life, the empty/low battery indicators **only blink in presence of KNX bus power**. In other words, it is once the bus recovers from the power failure that KCI 4 S0 will resume the battery level check.*

To get detailed information about the technical features of the device, as well as on the installation and security procedures, please refer to the corresponding **Datasheet**, bundled with the original package of the device and also available at [www.zennio.com](http://www.zennio.com)

## 2 CONFIGURATION

---

### 2.1 GENERAL

---

KCI 4 S0 incorporates four input channels for connecting up to four S0-pulse generators. Which of them will be in use is the first thing to configure.

On the other hand, two one-bit alarm objects will be available in the project by default. Their behaviour will be analogous to that of the on-board LED indicators that warn, respectively, about **low battery** and **empty battery**. While any of these LEDs remains on, the corresponding object will be sent to the bus with the value “1” every 15 minutes. Once the alarm situation is over, the LED turns off and the object is sent once with the value “0”.

Please, refer to section 1.2 for more details about the battery warnings.

#### ETS PARAMETERISATION

---

After importing the corresponding database in ETS and adding the device into the topology of the desired project, the configuration process begins by right-clicking into the device and selecting *Edit parameters*.

The “General” parameter screen is the only shown by default.

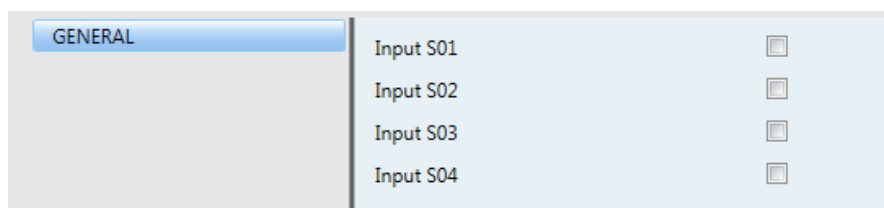


Figure 2. General.

Four checkboxes (one per input channel) let the integrator select which input channels will be in use. Additional entries will be included in the tab menu on the left depending on the number of active inputs.

The options for each of the inputs are detailed in the next sections.

On the other hand, the following two objects (which have their “R” flags active, so they can be read anytime) are available in the project topology at any time:

- **Alarm: Low Battery:** one-bit object that will send a “1” every 15 minutes if the battery level is low, and a “0” (only once) if this no longer happens. This object is analogous to the low-battery LED indicator.
- **Alarm: Empty Battery:** one-bit object that will send a “1” every 15 minutes if the battery level is critically low (or zero), and a “0” (only once) if this no longer happens. This object is analogous to the empty-battery LED indicator.

As with the LED indicators, the two alarm objects are mutually exclusive. Once the level of the battery becomes empty, KCI 4 S0 will start sending the empty-battery object and will interrupt the sending of the low-battery object (which will be then sent with the value “0” –only once–).

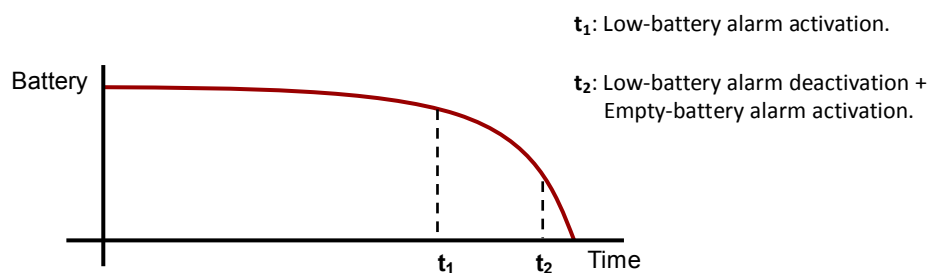


Figure 3. Battery Alarms

## 2.2 INPUT S0x

---

### 2.2.1 CONFIGURATION

---

Each input channel of KCI 4 S0 can be particularised for two different scenarios, depending on the type of the S0-pulse generator connected to it:

- Electric energy consumption meters.
- Water or gas volume consumption meters.

The available options will slightly differ for each case, although are entirely analogous. Although the compatibility is not guaranteed, KCI 4 S0 may also recognise pulses of other kind (even non-S0); however, previous testing is highly advisable in these cases.

As electric power, water or gas are being consumed, the pulse generator will send pulses to KCI 4 S0. To interpret the pulses, KCI 4 S0 needs to know the **pulse frequency**, which is the ratio (provided by the manufacturer) between the number of pulses sent by the external meter and the amount of electric energy (or water or gas) measured, i.e., the number of pulses sent per unit energy (or per unit volume). The higher the pulse frequency, the lower the consumption increase assumed by KCI 4 S0 on the reception of each pulse.

Regarding the measurement of **electric energy**, some more options and concepts are involved:

- The **initial energy value**, which is typically zero, can be customised (in parameters) so that KCI 4 S0 can resume a previous measurement. It is also possible to overwrite the current **energy value** anytime through a specific object.
- The value of the total energy accumulated can be sent to the KNX bus in terms of **Wh** (watt-hour) **kWh** (kilowatt-hour) or both, in the following situations:
  - **Cyclically**: after a certain time (configurable), the current value will be sent to the KNX bus.



**Note:** *the value sent by KCI 4 S0 is not strictly the instantaneous value, but the best known value at the moment. Depending on the pulse frequency, there may be a certain delay between the value being measured by the external meter and that corresponding to the last pulse.*

- **On value change:** the current value will be sent to the bus as soon as it is found to differ from the last sent by more than a certain amount (configurable). In other words, on the arrival of a pulse, the device will verify if the difference between the new value and the last sent is greater than the configured threshold; if it is, it will notify the KNX bus. It is also possible to set a *minimum time between sendings*, to avoid sending a new value too shortly after having sent the previous one.
- **Cyclical and on value change:** a combination of the above two options. The current value will be sent periodically, and also when it is found to differ from the last one sent by more than the specified amount. Note that the cyclical sending will be also conditioned by the *minimum time between sendings* (if configured), so that after a sending due to a value change there will not be cyclical sendings until the minimum time expires.
- Based on the accumulated energy consumption and the time lapse between the successive increments, KCI 4 S0 can obtain (and sent to the KNX bus) the corresponding **electric power**, in terms of watts or kilowatts.

The integrator can select one out of two **alternative methods** to calculate the electric power, named “**instantaneous**” and “**average**”. How each of them works is described in ANNEX I: Power Calculation.

Depending on the selected calculation method, different possibilities to send the value to the bus are available:

- Under “**instantaneous**”, it can be sent **cyclically**, after a **change greater than a certain value**, or in **both** cases (note that under “instantaneous” the power is re-calculated on the reception of every new pulse; therefore, the option to insert extra sendings –due to a value change– throughout the sending cycle does make sense).
- Under “**average**”, it can be sent **cyclically** or **cyclically conditioned to a change greater than a certain value**. Note that under “average” the

power is not re-calculated on the reception of a new pulse, but only before sending the value to the KNX bus. Therefore, only a cyclical sending (conditioned or not to a certain value change) makes sense here.

**Note:** *bear in mind that the electric power sent to the bus is always an estimate. Depending on the pulse frequency, even if there is some consumption, a power of 0 W may be estimated if no pulses have been received since the last sending. This situation is more likely to take place if the device is configured to send the power cyclically and after a certain change, as this causes sending the value more frequently to the bus.*

Analogously, when measuring **water or gas volumes**, the following options and concepts apply:

- There will be a parameter to set the **initial volume value** (typically zero) so that KCI 4 S0 can synchronise with other devices already monitoring the water or gas volume. The **current volume value** can also be overwritten anytime by writing to the corresponding object.
- As for the energy, the value of the water/gas volume can be sent to the KNX bus **cyclically**, after a certain **value change** or in both cases. This time, the value is expressed in **m<sup>3</sup>** (cubic metres, i.e., a thousand litres).
- Basing on the volume increment between two instants, KCI 4 S0 can calculate the **water or gas flow rate** and send it to the bus (in terms of m<sup>3</sup> per second or of litres per hour, according to the KNX standard) according to the parameterisation, which is entirely analogous to that for sending the electric power.

As for the electric power this is only an estimate, and can be obtained by two methods: “**instantaneous**” and “**average**” (see ANNEX I: Power Calculation).

## ETS PARAMETERISATION

---

For every active input, ETS shows one primary tab, containing another one named “Configuration”. The parameters in the latter are:

GENERAL

Input S01

Configuration

Type of Meter: Electricity

Pulses per Unit Energy: 1000 puls/kWh

ENERGY

Initial Energy Value: 0 Wh

Objects to Set the Energy Value: ☐

Sending Mode: Disabled

POWER

Power Calculation Method: Instantaneous (using the last two pulses received)

Sending Mode: Disabled

Figure 4. Input S0x – Configuration.

- **Type of Meter:** “Electricity” (default option) or “Water or gas”, depending on what the S0-pulse generator will be measuring.

All the remaining parameters will depend on the above selection, although **they are analogous** in both cases:

### Electricity:

- **Pulses per Unit Energy:** sets the pulse frequency, in pulses per kilowatt-hour (pulse/kWh). The available range is 1 to 65535 (the default is “1000”).
- **ENERGY:**

ENERGY

Initial Energy Value: 0 Wh

Objects to Set the Energy Value: ☐

Sending Mode: Disabled

- **Initial Energy Value:** sets the initial energy (in watts-hour) in case KCI is required (after download) to start the count from a value other than zero. The available range is 0 Wh to 4,294,967,295 Wh (the default is “0”).
- **Objects to Set the Energy Value:** checkbox to show or hide two objects that permit overwriting the value of the accumulated energy consumption

anytime. These are “[S0x] Set Energy (Wh)” and “[S0x] Set Energy (kWh)”. Both are four-byte size although one expects the values in terms of watt-hour, and the other in terms of kilowatt-hour.

- **Sending Mode:** sets when the value of the accumulated energy should be sent to the bus (through the four-byte objects “[S0x] Energy (Wh)” and “[S0x] Energy (kWh)”, respectively in watts-hour and kilowatts-hour):
  - Disabled: never.
  - Cyclical: every certain time, as set in “**Cycle Time**”. The available range is 10 to 255 seconds, 1 to 255 minutes and 1 to 255 hours.

Sending Mode	Cyclical
Cycle Time	1
	min

- On Value Change: whenever an increase is detected (over the last value sent) greater than a certain threshold, set in “**Minimum Energy Change to Send**”. The available range is 0 to 65535 tens of Wh.

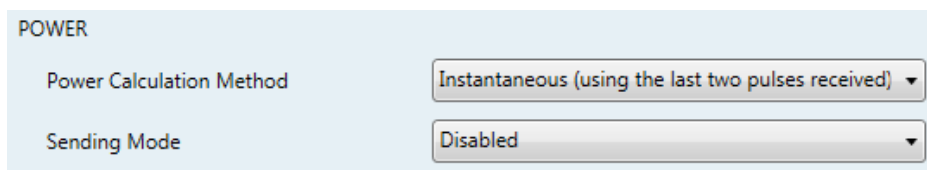
It is possible to impose (through parameter “**Minimum Time between Sendings**”) a minimum spacing between consecutive sendings. This will avoid sending values too often, even if the condition of the minimum change is met. The available range is 10 to 255 seconds.

Sending Mode	On Value Change
Minimum Time between Sendings	10 s
Minimum Energy Change to Send	1 x 10 Wh

- **Cyclical and on Value Change:** a combination of the above two.

Sending Mode	Cyclical and on Value Change
Cycle Time	1
	min
Minimum Time between Sendings	10 s
Minimum Energy Change to Send	1 x 10 Wh

## ● POWER:



POWER

Power Calculation Method: Instantaneous (using the last two pulses received) ▼

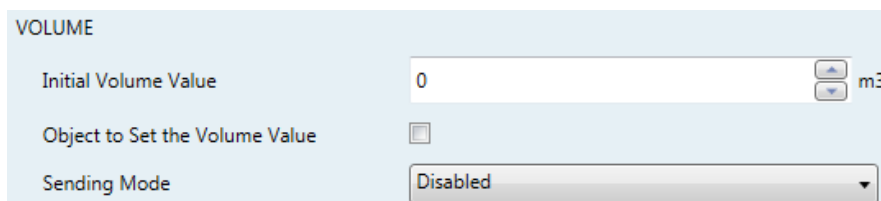
Sending Mode: Disabled ▼

- **Power Calculation Method:** “Instantaneous (using the last two pulses received)” (default option) or “Average (average value since the last sending)”. Please refer to ANNEX I: Power Calculation for details.
- **Sending Mode:** sets when the value of the power should be sent to the bus (through “[S0x] Power (4 bytes)” and “[S0x] Power (2 bytes)”, in watts and kilowatts respectively). The options are:
  - **For *instantaneous power calculation*:** “Disabled”, “Cyclical”, “On Value Change” and “Cyclical and On Value Change”. All of them are analogous to those for the energy sending (see above), although the units to set the minimum value change are here tens of watts instead of tens of watt-hours.
  - **For *average power calculation*:** “Disabled”, “Cyclical” and “Cyclical with Minimum Value Change”. The first two are analogous to the homonymous options for the energy sending (see above); the latter conditions the cyclical sending to a minimum difference over the last value sent, i.e., it is possible to set (in terms of watts) how much the new value must differ from the last one sent in order to actually sending it.

## Water or gas:

- **Pulses per Unit Volume:** sets the pulse frequency, in pulses per cubic metre (pulse/m<sup>3</sup>). The available range is 1 to 65535 (the default is “1000”).

## ● VOLUME:



VOLUME

Initial Volume Value: 0 m3

Object to Set the Volume Value: ☐

Sending Mode: Disabled ▼

- **Initial Volume Value:** sets the initial volume (in cubic metres), in case KCI is required (after download) to start the count from a value other than zero. The available range is 0 m<sup>3</sup> to 4,294,967,295 m<sup>3</sup> (the default is “0”).
- **Object to Set the Volume Value:** checkbox to show or hide a four-byte object (“[S0x] Set Volume”) that permits overwriting the value of the accumulated volume anytime. The value must be expressed in cubic metres.
- **Sending Mode:** sets when the value of the accumulated volume should be sent to the bus (through the four-byte object “[S0x] Volume”), expressed in cubic metres:
  - Disabled: never.
  - Cyclical: every certain time, as set in “**Cycle Time**”. The available range is 10 to 255 seconds, 1 to 255 minutes and 1 to 255 hours.

The screenshot shows a configuration window with a light blue background. On the left, there are two labels: 'Sending Mode' and 'Cycle Time'. To the right of 'Sending Mode' is a dropdown menu with 'Cyclical' selected. To the right of 'Cycle Time' is a numeric input field containing '1', with up and down arrow buttons on its right side. Below the numeric field is another dropdown menu with 'min' selected.

- On Value Change: whenever an increase is detected (over the last known value) greater than a certain threshold, set in “**Minimum Energy Change to Send**”. The available range is 0 to 65535 hundredths of a cubic metre (i.e., 0 to 655350 litres).

It is possible to impose (through the parameter “**Minimum Time between Sendings**”) a minimum spacing between consecutive sendings. This avoids sending values too often, even if the condition of the minimum change is met. The available range is 10 to 255 seconds.

The screenshot shows a configuration window with a light blue background. On the left, there are three labels: 'Sending Mode', 'Minimum Time between Sendings', and 'Minimum Volume Change to Send'. To the right of 'Sending Mode' is a dropdown menu with 'On Value Change' selected. To the right of 'Minimum Time between Sendings' is a numeric input field containing '10', with up and down arrow buttons on its right side, followed by a 's' unit indicator. To the right of 'Minimum Volume Change to Send' is a numeric input field containing '0', with up and down arrow buttons on its right side, followed by a 'x 0.01 m3' unit indicator.

- **Cyclical and on Value Change:** a combination of the above two.

Sending Mode	Cyclical and on Value Change
Cycle Time	1
	min
Minimum Time between Sendings	10 s
Minimum Volume Change to Send	0 x 0.01 m3

## ● FLOW:

FLOW	
Flow Calculation Method	Instantaneous (using the last two pulses received)
Sending Mode	Disabled

- **Flow Calculation Method:** “Instantaneous (using the last two pulses received)” (default option) or “Average (average value since the last sending)”. Please refer to ANNEX I: Power Calculation for details.
- **Sending Mode:** sets when the value of the flow should be sent to the bus (through “[S0x] Flow (4 bytes)” and “[S0x] Flow (2 bytes)”, in cubic metres per second and litres per hour respectively).
  - **For instantaneous flow calculation:** “Disabled”, “Cyclical”, “On Value Change” and “Cyclical and On Value Change”. All of them are analogous to those for the volume sending (see above), although the units to set the minimum value change are here tens of litres per hour –i.e., tenths of a cubic metre per hour– instead of tens of litres –i.e., tenths of a cubic metre–.
  - **For average flow calculation:** “Disabled”, “Cyclical” and “Cyclical with Minimum Value Change”. The first two are analogous to the homonymous options for the volume sending (see above); the latter conditions the cyclical sending to a minimum difference over the last value sent, i.e., it is possible to set (in tens of litres per hour –i.e., tenths of a cubic metre per hour–) how much the new value must differ from the last one sent in order to actually sending it.

## 2.3 INITIALISATION AND POWER LOSS

---

After an ETS download (as long as bus power is present):

- The **initial accumulated value** of the energy or volume will be the one set in parameters (zero by default).
- The value of the **power and flow** objects will be zero until they can be calculated (i.e., until KCI 4 S0 receives the pulses required for calculation).
- The **battery charge level** will start being monitored, so the associated LED indicators and alarm objects will be triggered when corresponding.

When a KNX bus power loss takes place:

- The **low-battery and empty-battery LED indicators will remain switched off** to extend the life of the battery. KCI 4 S0 will start again informing about the battery level after the recovery of the bus.
- In the absence of bus power and while there is battery charge enough, KCI 4 S0 **will still count any pulses** arriving from the pulse generator.

**Note:** *if the battery results to be discharged, these pulses will be lost. Therefore, after the recovery of the KNX bus, KCI 4 S0 will not take into account the corresponding energy or volume increments.*

Once the KNX bus power is recovered:

- The **low-battery or empty-battery LED indicator will start blinking** if the battery level is respectively low or very low. In such case, the corresponding communication object will be sent with the value "1". On the contrary, if the battery level is high enough, then the LEDs will stay off and the objects will remain with value "0".
- Any **timers** (cyclical sending, separation of consecutive values) will be reset to zero.
- The energy and volume objects will recover their previous values, being then **increased according to the increments received during the absence of**



**bus power** (provided that the battery level allowed to count them) and sent to the bus in case of having activated the option to send them on a value change (if the increment is greater than the minimum change parameterised).

- The value of the **power and flow** objects will be zero until they can be calculated again (i.e., until KCI 4 S0 receives the pulses required for calculation).

## ANNEX I: POWER CALCULATION

KCI 4 S0 can dynamically obtain from the pulses received the value of the electric power or the water / gas flow rate.

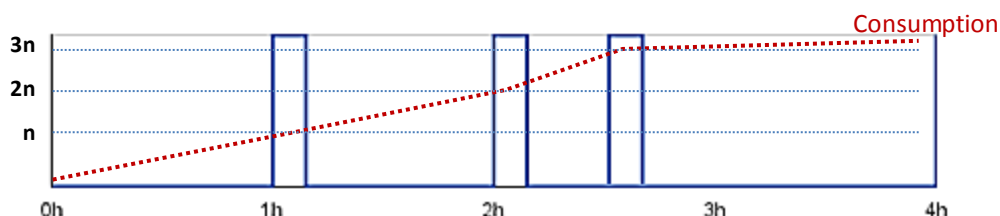


Figure 5. Pulse Reception.

The scenario shown in Figure 5 represents a sample case: KCI 4 S0 will be receiving discrete pulses at certain instants, as the energy, water or gas consumption increases. Assuming “n” energy (or volume) units per pulse (that is, a pulse frequency of “1/n”) and that the power / flow estimation is required to be re-calculated at the end of every hour, it is possible to proceed the following:

- Before receiving any pulses, the **accumulated energy** (or volume) is assumed to be zero (or the parameterised initial value), which is the *last known* value. However, this does not necessarily mean that there is absolutely no consumption at this time.
- After receiving the **first pulse**, the accumulated energy (or volume) is assumed to become “n”. Although that value is only true when the pulse arrives, during the second hour “n” will remain as the last known value.
- On the other hand, after the second hour, it is possible to estimate that there has been a **power** (or flow rate) of one pulse per hour (i.e., of “n” units per hour), as only one pulse was received between instants  $t = 1h$  and  $t = 2h$ .
  - Analogously, at the end of the **third hour**, one possible estimation would be that during the last hour there has been a power of two pulses per hour (2n units/hour), as two pulses arrived between  $t = 2h$  and  $t = 3h$ .
  - At  $t = 4h$ , as no pulses have been received since  $t = 3h$ , the **power estimation** could be 0 units/hour. However, in practice this does not mean

that there was no energy consumption, but that it was low enough as to not receive pulses for the moment, given the pulse frequency of the meter.

Note that the lower the power calculation intervals (and the higher the pulse frequency), the **higher the accuracy** of these results. In other words, to obtain the instant power value (and not an approximation) it would be necessary to take infinitesimal intervals and to have a continuous (non-discrete) reception of consumption increments.

Said that, KCI 4 S0 offers **two methods** for the power calculation, although both are based on a similar approach:

- Average value since the last sending:

This one the method used in the previous example.

Whenever a new cyclical sending of the power or flow value is required, KCI 4 S0 will re-calculate the power taking into account the time interval and the number of pulses received **since the last sending**.

$$\text{Power} = \Delta_{\text{Energy since the last sending}} / \Delta_{\text{time since the last sending}}$$

$$\text{Flow} = \Delta_{\text{Volume since the last sending}} / \Delta_{\text{time since the last sending}}$$

In this case, as the power is only re-calculated after specific intervals, it only makes sense to send the value to the bus cyclically (subject, if desired, to a minimum change over the last value sent), and not on the detection of a certain value change after the reception of a new pulse.

- Instantaneous value between the last two pulses:

The result is also an average value, however it is the most accurate approach to the actual instant power that KCI 4 S0 can achieve, as this time the value is re-calculated **whenever a new pulse arrives**. If no pulses have been received since the last sending (and, therefore, the power has not been re-calculated), the power will be considered zero if a new cyclical sending is required.

$$\text{Power} = \Delta_{\text{Energy due to a single pulse}} / \Delta_{\text{time since the last pulse}}$$

$$\text{Flow} = \Delta_{\text{Volume due to a single pulse}} / \Delta_{\text{time since the last pulse}}$$

This method allows sending the value to the bus cyclically, on a value change or in both cases.

**Example:** *Power Calculation Methods.*

*In Figure 5, assuming that the power has been parameterised to be sent to the bus only cyclically every two hours, the value at instant  $t = 2h$  will depend on the calculation method. For “**average**”, the estimated power at  $t = 2h$  will be “ $n / 2$ ” (one pulse divided by two hours); on the other hand, for “**instantaneous**”, the estimated power at  $t = 2h$  will be “ $n$ ” (one pulse divided by one hour).*

## ANNEX II. COMMUNICATION OBJECTS

- “**Functional range**” shows the values that, with independence of any other values permitted by the bus according to the object size, may be of any use or have a particular meaning because of the specifications or restrictions from both the KNX standard or the application program itself.
- “**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 to be assigned the corresponding updated value in case they depend on external elements. 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 Boot	P	Reboot	P	Name	Function
1, 11, 21, 31	4 Bytes	O	CTR--	DPT_ActiveEnergy	-2147483648 - 2147483647	-	✓	-		[S0x] Energy (Wh)	Electrical Energy (Wh)
2, 12, 22, 32	4 Bytes	O	CTR--	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-	✓	-		[S0x] Energy (kWh)	Electrical Energy (kWh)
3, 13, 23, 33	4 Bytes	I	C--W-	DPT_ActiveEnergy	-2147483648 - 2147483647	-		-		[S0x] Set Energy (Wh)	Set Energy Value in Wh
4, 14, 24, 34	4 Bytes	I	C--W-	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	-		-		[S0x] Set Energy (kWh)	Set Energy Value in kWh
5, 15, 25, 35	4 Bytes		CT---	DPT_Value_Power	-2147483648 - 2147483647	-		-		[S0x] Power (4 bytes)	Electrical Power (1 W resolution)
6, 16, 26, 36	2 Bytes		CT---	DPT_Power	-671088.64 - 670760.96	-		-		[S0x] Power (2 bytes)	Electrical Power (10 W resolution)
7, 17, 27, 37	4 Bytes	O	CTR--	DPT_Value_Volume	-671088.64 - 670760.96	-	✓	-		[S0x] Volume	Volume Consumption (m3)
8, 18, 28, 38	4 Bytes	I	C--W-	DPT_Value_Volume	-671088.64 - 670760.96	-		-		[S0x] Set Volume	Set Volume Value in m3
9, 19, 29, 39	4 Bytes		CT---	DPT_Value_Volume_Flux	-2147483648 - 2147483647	-		-		[S0x] Flow (4 bytes)	Volume Flow Rate (m3/s)
10, 20, 30, 40	2 Bytes		CT---	DPT_Value_Volume_Flow	-671088.64 - 670760.96	-		-		[S0x] Flow (2 bytes)	Volume Flow Rate (l/h)
41	1 Bit	O	CTR--	DPT_Alarm	0/1	-	W	-	W	Alarm: Low Battery	0=Normal; 1=Low Battery
42	1 Bit	O	CTR--	DPT_Alarm	0/1	-	W	-	W	Alarm: Empty Battery	0=Normal; 1=Empty Battery

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