



# HeatingBOX 230V 4X / 8X

## 4-Output / 8-Output 230V Heating Actuator

**ZCL-4HT230**  
**ZCL-8HT230**

Application programme version: [1.1]  
User manual edition: [1.1]\_a

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

# CONTENTS

---

Contents .....	2
Document Updates .....	3
1 Introduction .....	4
1.1 HeatingBOX 230V 4X and HeatingBOX 230V 8X .....	4
1.2 Installation.....	5
1.3 Start-Up and Power Loss.....	6
2 Configuration.....	7
2.1 General.....	7
2.2 Thermostats .....	8
2.3 Logic Functions.....	9
2.4 Heating Controls.....	10
2.4.1 Heating Control n.....	13
2.4.1.1 Configuration.....	13
2.4.1.2 Alarms.....	17
2.4.1.3 Cyclical Monitoring.....	19
2.4.1.4 Startup .....	21
2.4.1.5 Characteristic Curve.....	22
2.5 Manual Control .....	26
ANNEX I. Communication Objects.....	29

## DOCUMENT UPDATES

---

Version	Changes	Page(s)
[1.1]_a	<b>Changes in the application programme:</b> <ul style="list-style-type: none"><li>• Optimisation of the start-up process.</li></ul>	-

# 1 INTRODUCTION

---

## 1.1 HeatingBOX 230V 4X and HeatingBOX 230V 8X

---

**HeatingBOX 230V 4X** and **HeatingBOX 230V 8X** from Zennio are two heating-specific KNX actuators equipped, respectively, with four and eight independent outputs for controlling 230V valves.

The most outstanding features are:

- **4 or 8 configurable outputs** for controlling electromechanical valves.
- **A sole 230 Vac power input** for all outputs.
- **4 or 8 independent thermostats** (one per output).
- **10** customisable, multi-operation **logic functions**.
- **Manual operation / supervision** of the outputs through the on-board pushbuttons and LEDs.

## 1.2 INSTALLATION

HeatingBOX 230V 4X / 8X 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 programme may be downloaded.

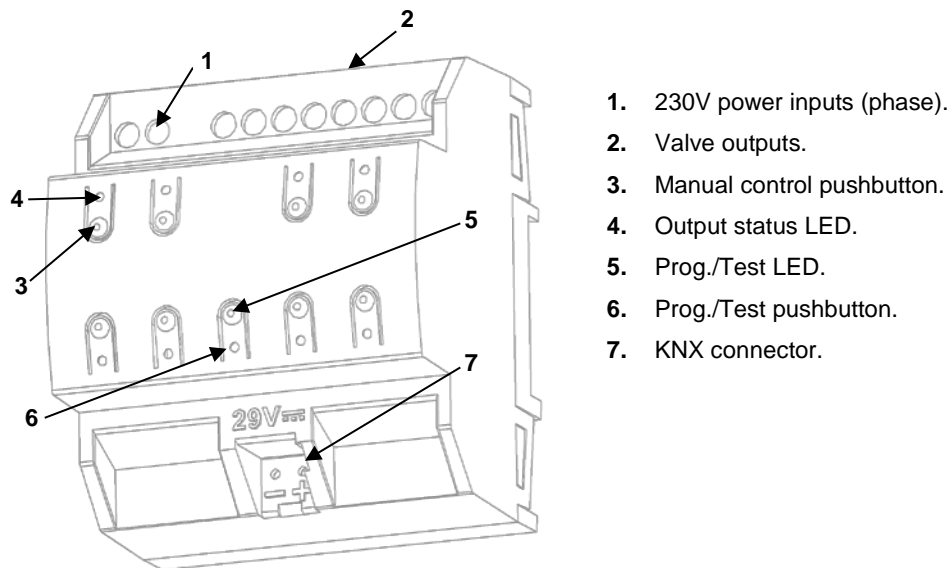


Figure 1. HeatingBOX 230V 8X.

**Note:** the above element diagram is slightly different for HeatingBOX 230V 4X, although entirely analogous.

The main elements of the device are described next.

- **Test/Prog. Pushbutton (6):** 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 every 0.5 seconds.

- **Phase Inputs (1):** slots for the connection of the voltage lines (phase). Both inputs are internally shorted.

- **Outputs (2):** output ports for the insertion of the stripped cables of the systems being controlled by the actuator. Please secure the connection by means of the on-board screws.
- **Manual control pushbuttons (3):** pushbuttons for a direct control of the outputs during the set-up process. See section 2.5.

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

### 1.3 START-UP AND POWER LOSS

---

During the start-up of the device and depending on the configuration, some specific actions may be performed. For example, the integrator can set whether the outputs should change to a particular position and whether the device should send certain objects to the bus after the power recovery. Please consult the next sections of this document for further details.

On the other hand, when a bus power failure takes place, the device will interrupt any pending actions, and will save its state so it can be recovered once the power supply is restored.

## 2 CONFIGURATION

---

### 2.1 GENERAL

---

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*.

#### ETS PARAMETERISATION

---

From the General screen, available by default, it is possible to activate/deactivate all the required functionality.

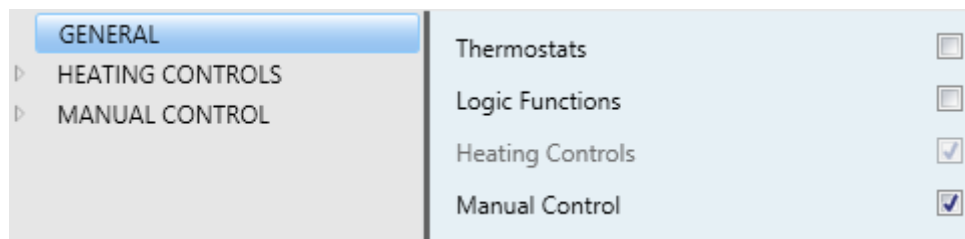


Figure 2. General screen

- Once activated, **Logic Functions**, **Thermostats**, **Heating Controls** and **Manual Control** bring additional tabs to the menu on the left. These functions and their parameters will be explained in later sections of this document.

**Manual Control** and **Heating Controls** are enabled by default, and so are their corresponding configuration tabs.

## 2.2 THERMOSTATS

---

HeatingBOX 230V 4X and HeatingBOX 230V 8X implement **four and eight Zennio thermostats**, respectively, which can be enabled and configured independently.

Please refer to the specific “**Zennio Thermostat**” user manual (available in the HeatingBOX 230V 4X / 8X product section at the Zennio homepage, [www.zennio.com](http://www.zennio.com)) for detailed information about the functionality and the configuration of the related parameters.



## 2.3 LOGIC FUNCTIONS

---

This module makes it possible to perform numeric and binary operations to incoming values received from the KNX bus, and to send the results through other communication objects specifically enabled for this purpose.

HeatingBOX 230V 4X / 8X can implement **up to 10 different and independent functions**, each of them entirely customisable and consisting in **up to 4 consecutive operations each one**.

The execution of each function can depend on a configurable **condition**, which will be evaluated every time the function is **triggered** through specific, parameterisable communication objects. The result after executing the operations of the function can also be evaluated according to certain **conditions** and afterwards sent (or not) to the KNX bus, which can be done every time the function is executed, periodically or only when the result differs from the last one.

Please refer to the specific “**Logic Functions**” user manual (available in the HeatingBOX 230V 4X / 8X product section at the Zennio homepage, [www.zennio.com](http://www.zennio.com)) for detailed information about the functionality and the configuration of the related parameters.

## 2.4 HEATING CONTROLS

---

The HeatingBOX 230V 4X / 8X actuator incorporates **four or eight outputs**, each of them capable of controlling one or several heating valves at 230V.

The actuator provides a set of common options to configure the general operation of these valves:

- A **start-up delay**, to prevent performing control orders over the outputs immediately after the device initialisation. Note that this start-up delay is interrupted if alarm or lock orders (see section 2.4.1.1) are received.
- Short-circuit, overload and all-valves-closed **notifications**, which are periodically sent through specific one-bit objects in case such events occur:
  - **Short-circuit or overload**: a specific object is sent periodically in case a short circuit is detected on a particular output. On the other hand, overload detection is performed jointly for outputs 1-4 and 5-8 (one object will be available per group). In case of any of the two errors, the current flow through the corresponding output or group of outputs will be interrupted (the valve will return to idle position), until a new control order is received.
  - **All valves closed**: it may be significant for the climate system whether all valves are currently closed or not (e.g., to switch off the climate unit in case they are). This notification can be restricted for **a specific set of valves** (so the state of the other valves is not taken into account) and also **delayed**, to ensure that such situation is not transitory and persists in time.
- **Maximum control value** object, which reports (with a configurable periodicity) the maximum of the current control values of the outputs that may have been marked for such purpose. An **external** value (i.e., received through a specific object) can be also considered for this comparison, which brings the option to link multiple actuators together and to determine the maximum of all of them.
- The optional **anti-seize** protection function, which automatically switches the state of those outputs that have been in the same position for more than a configurable time. After a certain duration (also configurable), they return to their previous state.

## ETS PARAMETERISATION

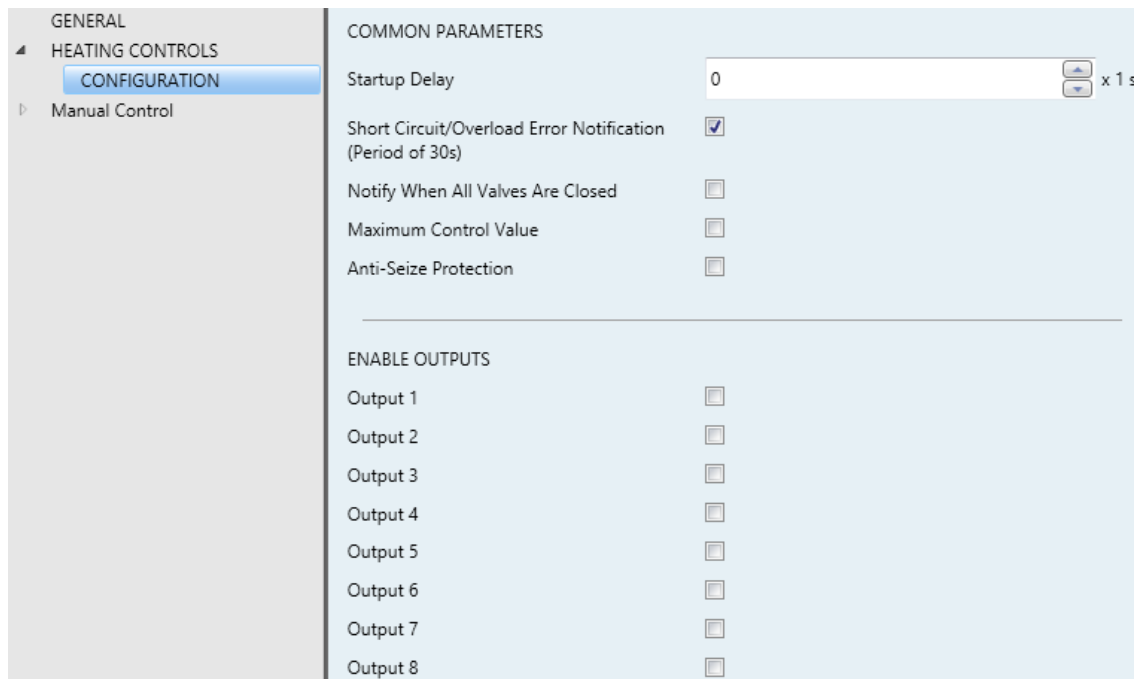


Figure 3. Heating Controls – Configuration (HeatingBOX 230V 8X).

From the “Configuration” screen is possible to enable/disable all the common functions required. In addition, each output can be individually enabled.

- **Startup Delay:** delay (0-255 seconds) after the device initialisation.
- **Shortcircuit/Overload Error Notification:** when enabled, the one-bit objects “[HCx] Shortcircuit” (one per output) and “[HCx-y] Overload” (one for outputs 1-4 and one for outputs 5-8; the latter only in HeatingBOX 230V 8X) will become available. These objects will be sent to the bus with the value ‘1’ every 30 seconds as long as the error persists, and with the value ‘0’ (once) when the error is over.
- **Notify When All Valves Are Closed:** if enabled, it provides the one-bit object “[HC] All Valves are Closed”, which will be sent to the bus with a configurable value, depending on whether are valves are found to be closed or not. A **delay** can be parameterised to ensure the persistence of this situation.

**Note:** *this function ignores those outputs that have not been specifically marked for this purpose (see section 2.4.1).*

Figure 4. Notify when all valves are closed.

- **Value:** “Send 0” or “Send 1”.
- **Delay:** 0 to 1440 minutes; 0 to 24 hours.

**Note:** *this delay must be greater than the cycle time (either in case of making use of an external device or of the 1-byte control method). See section 2.4.1.*

- **Maximum Control Value:** enables the 1-byte object “[HC] Max. Control Value (Output)”, which reports the maximum of the control values of the outputs marked for this purpose (see section 2.4.1).

Figure 5. Maximum control value.

- **Add External Value to Calculation:** enables the 1-byte object “[HC] Max. Control Value (Input)” for the reception of an additional value which will be taken into account to determine the above maximum.
- **Resending Period:** 0-255 minutes; 30 by default.

**Note:** *this function ignores those outputs that have not been specifically **marked** for this purpose (see section 2.4.1).*

- **Anti-Seize Protection:** enables or disables the anti-seize protection function. The “[HCx] Anti-Seize Protection” binary object will send the value “1” when the anti-seize process is triggered for output “x”, and the value “0” when it is over.

Anti-Seize Protection	<input checked="" type="checkbox"/>
Periodicity	7 <input type="text"/> x 1 day
Duration	5 <input type="text"/> x 1 min

Figure 6. Heating Controls – Anti-Seize Protection.

- **Periodicity:** sets the maximum time an output can remain still (1-30 days; 7 by default).
- **Duration:** sets for how much time the output should switch its position, in case it has been still for more than the above periodicity (1 to 255 minutes; 5 by default).
- **Output n:** enables or disables output number “n”. A specific set of pages is added to the tab tree on the left per enabled output. Please see section 2.4.1.

## 2.4.1 HEATING CONTROL n

---

Each output enabled provides a Configuration screen with a set of general parameters, as well as some more screens with optional settings.

### 2.4.1.1 CONFIGURATION

---

The general configuration of each output entails the following options:

- **Control method:** sets the type (binary or percentage) of the object that will receive the control orders for the output. These commands must be received from a thermostat (either internal or external), so the object type must match the control type performed by the thermostat.
 

**Important:** *percentage values are internally transformed into a PWM signal (which is what the actuator actually applies to the output), the period of which must be defined by the integrator, while under a one-bit control such period is assumed to be already defined in the thermostat. In both cases, to ensure a proper behaviour, it is important that the PWM period is not lower than the valve transition time (opening / closing).*
- **Valve type:** sets the type of the valve connected to the output, which may be **normally open** (in the absence of power, the valve remains open) or **normally**

**closed** (in the absence of power, the valve remains closed).

The format of the control orders is independent of the valve type. The range of the binary or percentage objects (depending on the control method) remains the same in any case – the actuator internally decides whether to apply power or not to the valve depending on the order received and the valve type.

The following table summarises the device behaviour according to the valve type and the control order received. Note that the status LED of each output indicates whether current is supplied to the output or not and not whether the valve is open or closed (it depends on the valve type).





Valve type	Control order		Output power	LED	Valve state
	1 bit	1 byte			
Normally closed	1	100%	Yes		Open
	0	0%	No		Closed
Normally open	1	100%	No		Open
	0	0%	Yes		Closed

Table 1 Device behaviour according to the valve type.

- Whether the output (valve) should be considered or not for the **notification in case all valves are closed** (see section 2.4).
- Activation and deactivation of the **status objects** of the one-bit and one-byte control values.
- **Lock** function, which provides a one-bit object to temporarily deactivate the control of the output (valve), being possible to leave it as is until it gets unlocked, or to switch it to a particular position.
- **Alarms**: see section 2.4.1.2.
- **Cyclical monitoring**: see section 2.4.1.3.
- **Characteristic curve**: see section 2.4.1.5.
- **Start-up type**: see section 2.4.1.4.

## ETS PARAMETERISATION

The configuration screen of each individual heating control contains the following parameters:

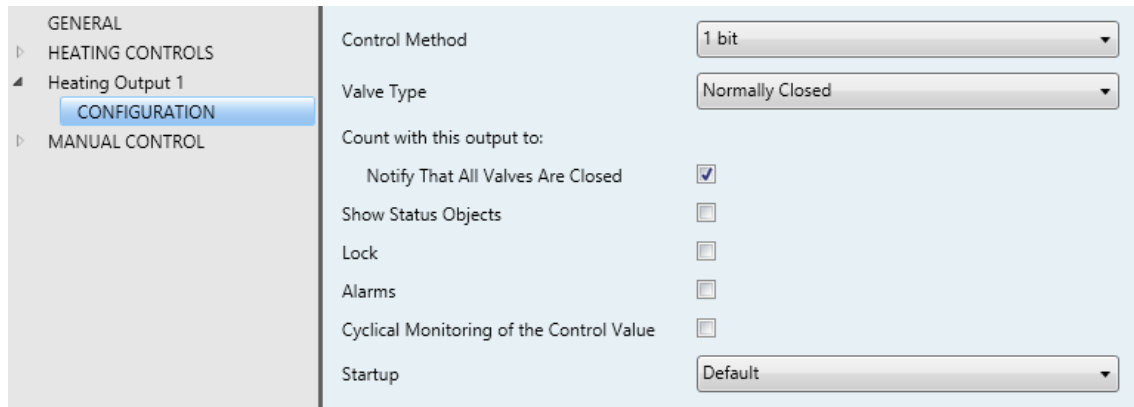


Figure 7. Heating Control – Configuration.

### • Control method:

- “1 bit”: the orders to close (value “0”) and to open (value “1”) the valve must be received through the one-bit object “[HCx] Control Value – 1 bit”.
- “1 byte”: the orders to close (value “0%”) and to open (value “100%”) the valve must be received through the one-byte object “[HCx] Control Value – 1 byte”. This method entails the following parameter as well:
  - **Cycle time**: sets the PWM cycle time, i.e., the base time for the PWM modulation. 1-3600 seconds or 1-1440 minutes.

The device will perform opening and closing orders to the valve alternatively in order to satisfy the control value.

**Example:** if the cycle time is 10 minutes and a control value of 20% is received, then (every ten minutes):

- The valve will remain open for 2 minutes.
- Afterwards it will remain closed for 8 minutes.

### • Valve type::

- “Normally Closed”.

- “Normally Open”.

**Note:** *the control value to close a valve is always “0” (under a one-bit control) or “0%” (under a 1-byte control) no matter if the valve is normally open or normally closed.*

- **Show Status Objects:** adds one or two (depending on the control type) status objects to the project in relation to the control value.

- **“[HCx] Control Value – 1 bit (Status)”** (available in both, the one-bit and the one-byte controls): it will send the value “0” when the output closes and the value “1” when it opens.

- **“[HCx] Control Value – 1 byte (Status)”** (only for the one-byte control): it will send the current control value whenever it changes.

**Note:** *this object corresponds to the input value of the characteristic curve (value in the X axis), not to the output value. See section 2.4.1.4*

- **Count with this output to...**

- **Notify That All Valves Are Closed:** when checked, the current output will be taken into account to determine whether all valves are closed or not.

**Note:** *this option is only available if the **Notify When All Valves Are Closed** function has been enabled (see section 2.4).*

- **Calculate the Max. Control Value:** when checked, the current output will be taken into account to calculate the maximum control value amongst all the outputs.

**Note:** *this option is only available under a **one-byte** control and when the **Maximum Control Value** function has been enabled (see section 2.4).*

- **Lock:** provides the **“[HCx] Lock”** one-bit object, which permits locking (with the value “1”) or unlocking (with the value “0”) the control of the output. While locked, the output ignores any control commands.

When this function has been enabled, the following parameter appears:

- **Action:** sets an action to be performed by the output when it gets locked.



- **One-bit control:** “No Change”, “Off”, “On”.
  - **One-byte control:** “No Change”, “Specific Control Value”. When selecting the latter, the parameter **Control Value** lets introducing the desired percentage value.
- **Alarms:** enables or disables the Alarm function. See section 2.4.1.1.
  - **Cyclical Monitoring of the Control Value:** enables or disables the Cyclical Monitoring function. See section 2.4.1.2.
  - **Characteristic Curve:** enables or disables the Characteristic Curve function. See section 2.4.1.4.
  - **Startup:** sets whether, during the device initialisation, the output should behave normally (“Default”) or perform a custom action (“Custom”). See section 2.4.1.3.

#### 2.4.1.2 ALARMS

---

The Alarms function allows changing the state of the output on the reception of an alarm trigger from the KNX bus. It is possible to configure the state the output will be switched to, both on the alarm **activation** and on the alarm **deactivation**.

**Cyclically monitoring** the alarm trigger is also possible by defining a certain time period. The actuator will check that the alarm or no-alarm state is received at least once before the period expires (note: this check doesn't take place unless the object has been received at least once). In case the object stops being refreshed (i.e., the actuator does not receive an updated value anymore), the alarm action will be performed as well, for safety reasons.

Regarding the deactivation of the alarm, it is also possible to configure a simple deactivation or an **acknowledgement-demanding** deactivation.

- The first case triggers the deactivation action as soon as the alarm object recovers its normal value.
- The second one, on the other hand, requires that an external acknowledgement (through another object) is received after the alarm object has recovered its normal value.

**Notes:**

- **Alarms always prevail** over other functions (e.g.: lock orders during the alarm state will be ignored; alarm orders during the lock state will not).
- **Alarm has priority over Alama 2.** If a channel is in alarm 2 and alarm 1 is activated, the alarm 1 action will be executed until the alarm 1 is deactivated, resuming the alarm 2 status (but without executing the alarm 2 action again). However, if a channel is in state of alarm 1 and alarm 2 is activated, it will be ignored and the channel will continue in a state of alarm 1, until the first alarm is deactivated when the channel will execute the alarm 2 action.

**ETS PARAMETERISATION**

After enabling “**Alarms**” in the Configuration screen (see section 2.4.1.1), a new tab will be incorporated into the tab tree on the left.

While the parameters of Alarm 1 are displayed by default, parameters of Alarm 2 are only shown after thick the corresponding checkbox.

Figure 8. Heating Control – Alarms.

- **Trigger:** sets the value (“0” –default– or “1”) that when received from the KNX bus through object “[HCx] Alarm” will be interpreted by the actuator as an alarm trigger and will therefore initiate the action configured below.

- **Cyclical Monitoring Period:** sets every how much time, at most, the alarm object should be updated from the bus after an initial reception has already taken place. If exceeded, and for safety reasons, the alarm action will be triggered as well. If this parameter is set to zero (default option), the cyclical monitoring function will remain disabled. Permitted values are 1 to 3600 seconds; 1 to 1440 minutes; and 1 to 24 hours.
- **Action:** sets the state the output should acquire once a lock order is received. The options depend on the control method:
  - **1 bit:** “No Change”, “Off”, “On”.
  - **1 byte:** “No Change”, “Specific Control Value”. When selecting the latter, the parameter **Control Value** shows up to enter the percentage value.
- **Deactivation – Mode:** “Normal” (default) or “Frozen (Acknowledgement Required)”. The second option enables a new 1-bit object, **“[HCx] Unfreeze Alarm”**, which should be used for externally unfreezing the alarm once **“[HCx] Alarm”** has received the no-alarm value (i.e., the inverse of the trigger value).

**Note:** *the acknowledgement should be sent necessarily after **“[HCx] Alarm”** has acquired the no-alarm value. Sending it while the trigger value is still active will have no effect.*
- **Deactivation – Action:** sets the state the output should acquire once the alarm has been deactivated (and acknowledged, if required). The options depend on the control method:
  - **1 bit:** “No Change”, “Off”, “On” or “Last Control Value (Before the Alarm)”.
  - **1 byte:** “No Change”, “Specific Control Value” or “Last Control Value (Before the Alarm)”. When selecting “Specific Control Value”, the parameter **Control Value** shows up to enter the percentage value.

#### 2.4.1.3 CYCLICAL MONITORING

It is possible to perform a **cyclical monitoring** of the control orders that are received from the external device (e.g.: a thermostat) that controls this module, with the aim of detecting communication issues.

This requires defining a **cyclical monitoring interval**, in other words, how frequently these orders are expected to arrive from the bus. If this period is exceeded with no orders received, a certain reaction will be performed and the error will be notified through a specific object.

The **reaction in case of failure** can consist in:

- Set the output status to a specific position.
- Doing nothing.

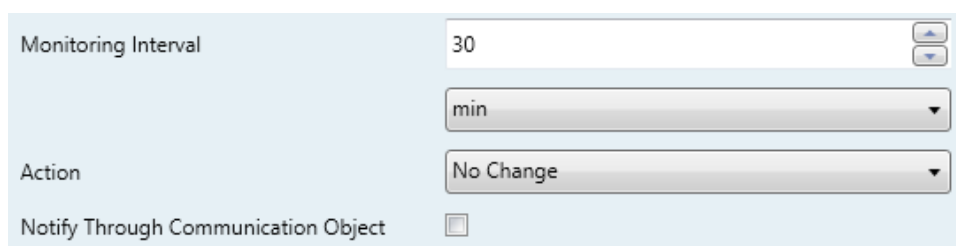
The **notification object** will send the value “1” when the error is detected and the value “0” once it has been solved, that is, once the external orders are resumed.

#### **Notes:**

- *Please bear in mind that the monitoring interval must be at least equal or greater than the PWM control period.*
- *This check doesn't take place unless the object has been received at least once.*

### **ETS PARAMETERISATION**

After enabling the cyclical monitoring function from the “Configuration” tab (section 2.4.1.1), the “Cyclical Monitoring” tab will be available, containing the following parameters.



Monitoring Interval	30
	min
Action	No Change
Notify Through Communication Object	<input type="checkbox"/>

**Figure 9.** Heating Control – Cyclical Monitoring.

- **Monitoring Interval:** either 1 to 3600 seconds, or 1 to 1440 minutes, or 1 to 24 hours.
- **Action:** state the output should acquire if the cyclical monitoring is interrupted. The options depend on the control method:

- **1 bit:** “No Change”, “Off”, “On”.
  - **1 byte:** “No Change”, “Specific Control Value”. When selecting the latter, the parameter **Control Value** shows up to enter the percentage value.
- **Notify Through Communication Object:** enables the “[HCx] Control Value - Error” binary object, which will send the value “1” in case the communication has been interrupted, and the value “0” once the error is over.

#### 2.4.1.4 STARTUP

The Startup function brings the option to set the output to a particular state at the start-up of the actuator, provided that there are no other active functions with a higher priority (alarms, locks, etc.). To that end, the integrator may select one of the following two options:

- **Default start-up:** after an ETS download the status of the output will be closed (values “0” and “0%” if the valve is normally closed, and “1” and “100%” if normally open), while after a bus power failure it will recover the last status.
- **Custom start-up:** the integrator may select whether the output is required to recover its previous status (on the very first start-up of the device, it will remain closed) or acquire a specific status. The status object can also be configured to be sent to the bus (with a customisable delay).

### ETS PARAMETERISATION

Only in case of opting for a **custom start-up** in the Configuration screen (see section 2.4.1.1), a new tab will be incorporated into the tab tree on the left.

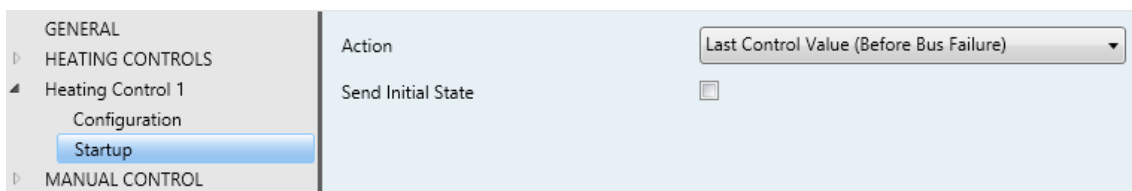


Figure 10 Heating Control - Startup

It contains the following parameters:

- **Action:** sets the state the output should acquire at the start-up of the actuator. The options depend on the control method:
  - **1 bit:** “Last Control Value (Before Bus Failure)”, “Off”, “On”.
  - **1 byte:** “Last Control Value (Before Bus Failure)”, “Specific Control Value”. When selecting the latter, the parameter **Control Value** shows up to enter the percentage value.
  
- **Send Initial State:** sets whether the status object of the output should be sent to the KNX bus (in order to update other KNX devices) after the start-up of the actuator. It is possible to impose a delay (0 to 3600 seconds; 0 to 1440 minutes; 0 to 24 hours) to ensure that this value is sent once the other devices are ready to receive it.

#### 2.4.1.5 CHARACTERISTIC CURVE

**Note:** *this function is only available under the one-byte control method (see section 2.4.1).*

This function allows the integrator to define a specific curve for the valve control.

By default, all control values received are directly applied to the valve (through a PWM signal) as shown in Figure 11.

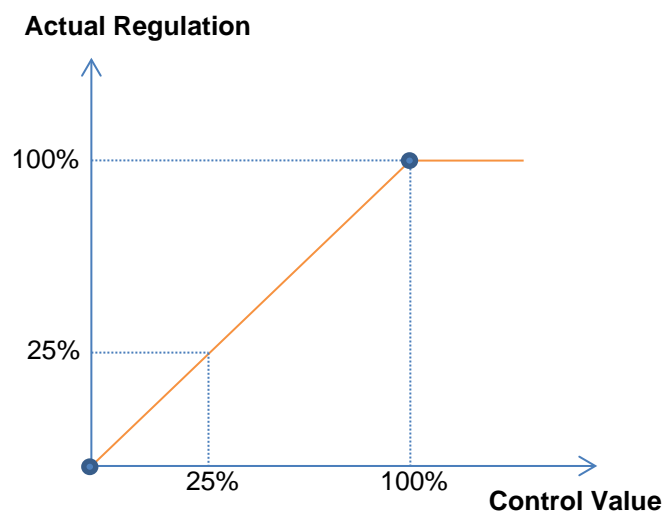


Figure 11. Default Control Curve.

For a more complex control, the actuator can implement a characteristic curve, defined through **three custom control points**, i.e., three pairs of values of the form (x, y). This lets the definition of custom curves, such as:

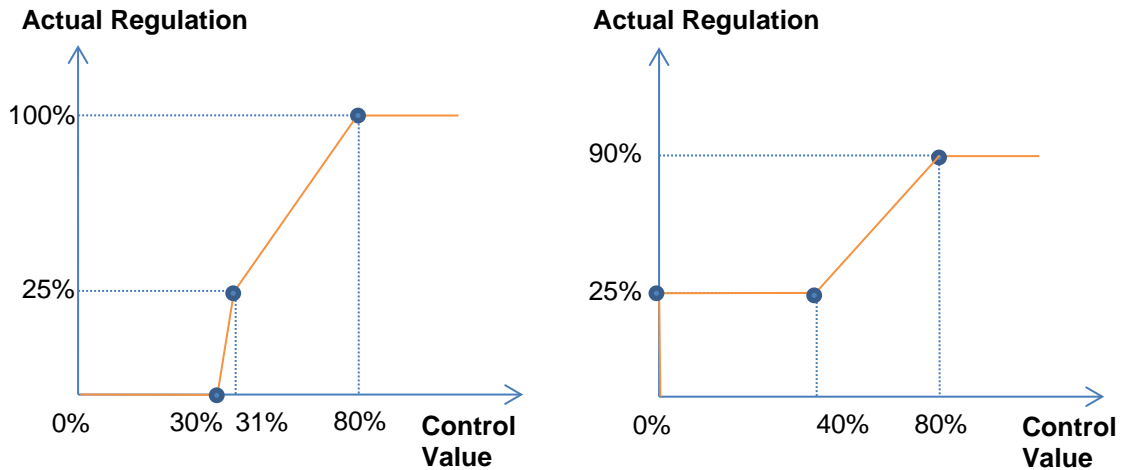


Figure 12. Custom Control Curves.

In case the first and last points do not correspond (respectively) to values 0% and 100%, the characteristic curve will be defined as follows (see Figure 12 and Figure 13):

- For **control values lower than that of the first point**, the actual regulation value will be 0%. For example, in Figure 13 all control values under 20% imply a regulation value of 0%.
- For **control values greater than that of the last point**, the actual regulation value will be 100%. For example, in Figure 13 all control values over 80% imply a regulation value of 100%.

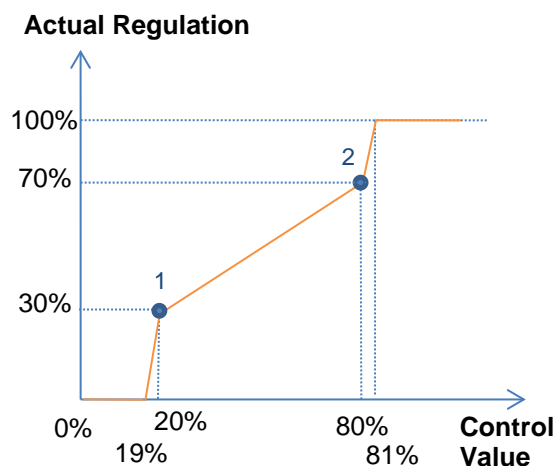


Figure 13. Characteristic Curve – Values Out of Range.

**Notes:**

- Please ensure that **different control values are assigned different regulation values**. Otherwise, the regulation may present issues.
- **Intermediate values** between the user-defined points are calculated by linear interpolation.
- The **status objects** will always refer to the control values, between 0% and 100%, with independence of the curve being implemented, which is typically irrelevant for the final user.

**PARAMETRIZACIÓN ETS**

After enabling “**Characteristic Curve**” in the Configuration screen (see section 2.4.1.1), a new tab will be incorporated into the tab tree on the left.

Value Pair	Control Value (%)	Valve Value (%)
Value Pair 1	0	0
Value Pair 2	100	100
Value Pair 3	0	0

**Figure 14.** Heating Control – Characteristic Curve.

Contains the parameters required to define up to three points of the characteristic curve:

- **Control Value (%)**: input value received through the one-byte control object (value in the “x” axis).
- **Valve Value (%)**: real value applied to the output to control the valve when the above control value is received (value in the “y” axis).



**Note:** please ensure that points 1 through 3 have different control values (i.e., each control value must correspond to only one valve value), and that these values are sorted in ascending order (control value 1 < control value 2 < control value 3). Otherwise, the output regulation function may behave abnormally.

## 2.5 MANUAL CONTROL

---

HeatingBOX 230V 4X / 8X allows controlling the outputs through the pushbuttons on the top of the device. Each output has its associated pushbutton.

Manual operation can be done in two different ways, named as **Test On Mode** (for testing purposes during the configuration of the device) and **Test Off Mode** (for a normal use, anytime). Whether both, only one or none of these modes are available needs to be parameterised in ETS. Moreover, it is possible to enable a specific binary object for locking and unlocking the manual control in runtime.

### Note:

- *The **Test Off mode** will be active (unless it has been disabled by parameter) after a download or a reset with no need of a specific activation – the pushbuttons will respond to user presses from the start.*
- *On the contrary, switching to the **Test On mode** (unless disabled by parameter) needs to be done by long-pressing the Prog./Test button (for at least three seconds), until the LED is no longer red and turns yellow. From that moment, once the button is released, the LED light will remain green to confirm that the device has switched from the Test Off mode to the Test On mode. After that, an additional press will turn the LED yellow and then off, once the button is released. This way, the device leaves the Test On mode. Note that it will also leave this mode if a bus power failure takes place.*
- *The manual control of the outputs is intended to perform on/off switches, independently of the method of control configured (one-bit or one-byte). In other words, it does allow testing the opening and closing of the valves, but intermediate positioning is not possible.*

### Test Off Mode

Under the Test Off Mode, the outputs can be controlled through both their communication objects and the actual pushbuttons located on the top of the device. Also, during this manual control mode, the control commands received by the bus will continue to be analysed and executed.

If the pushbutton on an **enabled output** is pressed, the output will behave as if an order to open or to close the valve had been received through the corresponding communication object, and will also send the status objects when required.

Regarding the lock and alarm functions, the device will behave under the Test Off mode as usual. Button presses during this mode are entirely analogous to the reception of the corresponding orders from the KNX bus.

## Test On Mode

After entering the Test On mode, it will only be possible to control the outputs through the on-board pushbuttons. Control orders received through communication objects will be ignored.

Short or long pressing the button will commute the on-off state of the output. The LED will light in green while the button is being hold.

The lock and alarm functions as well as any orders received from the KNX bus will not have an effect over the output status while the device is under the Test On mode. The status objects will not be sent to the bus, either.

On the contrary, the alarm and lock objects will be re-evaluated after leaving the Test On mode, so any changes that may have taken place in Test On will be considered when leaving.

**Important:** *the device is delivered from factory with both manual modes (Test Off and Test On) enabled by default.*

## ETS PARAMETERISATION

After enabling “**Manual Control**” in the Configuration screen (see section 2.1), a new tab will be incorporated into the tab tree on the left.

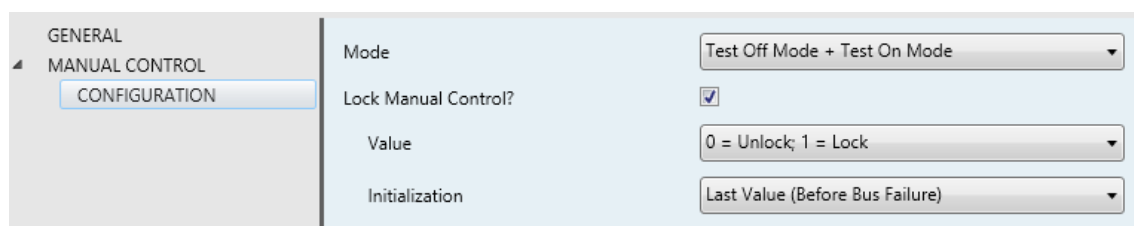


Figure 15. Manual Control.

The only two parameters are:

- **Mode:** options are “Disabled”, “Only with Test Mode Off”, “Only with Test Mode On” and “Test Off Mode Mode and Test On Mode” (default).

Depending on the selection, the device will permit using the manual control under the Test Off, the Test On, or both modes. Note that, as stated before, using the Test Off mode does not require any special action, while switching to the Test On mode does require long-pressing the Prog./Test button.

- **Lock Manual Control?:** unless the above parameter has been “Disabled”, the Lock Manual Control parameter provides an optional procedure for locking the manual control in runtime. When this checkbox is enabled, object “**Lock Manual Control**” turns visible, as well as two more parameters:

- **Value:** defines whether the manual control lock/unlock should take place respectively upon the reception (through the aforementioned object) of values “0” and “1”, or the opposite.
- **Initialization:** sets how the manual control should remain after the device start-up (after an ETS download or a bus power failure): “Unlocked”, “Locked” or “Last Value (Before Bus Failure)” (default; on the very first start-up, this will be unlocked).

## ANNEX I. 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 programme itself.

**Note:** objects corresponding to outputs 5-8 are not available in HeatingBOX 230V 4X.

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	Name	Function
1	1 Bit	I	C - - W -	DPT_Switch	0/1	Bloquear control manual	0 = Desbloquear; 1 = Bloquear
	1 Bit	I	C - - W -	DPT_Switch	0/1	Bloquear control manual	0 = Bloquear; 1 = Desbloquear
2	1 Byte	I	C - - W -	DPT_SceneControl	0-63; 128-191	[Termostato] Escenas: entrada	Valor de escena
3, 33, 63, 93, 123, 153, 183, 213	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Fuente de Temperatura 1	Temperatura de sensor externo
4	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Fuente de Temperatura 2	Temperatura de sensor externo
5	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Temperatura Efectiva	Temperatura efectiva de control
6	1 Byte	I	C - - W -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Modo Especial	Valor de modo de 1 byte
7	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Tx] Modo Especial: confort	0 = Nada; 1 = Disparo
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Modo Especial: confort	0 = Apagado; 1 = Encendido
8	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Tx] Modo Especial: standby	0 = Nada; 1 = Disparo
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Modo Especial: standby	0 = Apagado; 1 = Encendido
9	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Tx] Modo Especial: económico	0 = Nada; 1 = Disparo
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Modo Especial: económico	0 = Apagado; 1 = Encendido
10	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Tx] Modo Especial: protección	0 = Nada; 1 = Disparo
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Modo Especial: protección	0 = Apagado; 1 = Encendido
11	1 Bit	I	C - - W -	DPT_Window_Door	0/1	[Tx] Estado de Ventana (entrada)	0 = Cerrado; 1 = Abierto
12	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Tx] Prolongación de Confort	0 = Nada; 1 = Confort Temporizado
13	1 Byte	O	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Modo Especial (Estado)	Valor de modo de 1 byte
14	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Consigna	Consigna del termostato

	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Consigna Básica	Consigna de referencia
15	1 Bit	I	C - - W -	DPT_Step	0/1	[Tx] Consigna (Paso)	0 = -0.5°C; 1 = +0.5°C
16	2 Bytes	I	C - - W -	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Consigna (Offset)	Valor de coma flotante
17	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Consigna (Estado)	Consigna actual
18	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Consigna Básica (Estado)	Consigna básica actual
19	2 Bytes	O	C T R - -	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Consigna (Estado de Offset)	Valor actual del offset
20	1 Bit	I	C - - W -	DPT_Reset	0/1	[Tx] Reinicio de Consigna	Reinicio a valores por defecto
	1 Bit	I	C - - W -	DPT_Reset	0/1	[Tx] Reiniciar Offsets	Reiniciar offset
21	1 Bit	I	C - - W -	DPT_Heat_Cool	0/1	[Tx] Modo	0 = Enfriar; 1 = Calentar
22	1 Bit	O	C T R - -	DPT_Heat_Cool	0/1	[Tx] Modo (Estado)	0 = Enfriar; 1 = Calentar
23	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] On/Off	0 = Apagado; 1 = Encendido
24	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] On/Off (Estado)	0 = Apagado; 1 = Encendido
25	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Tx] Variable de Control (Enfriar)	Control PI (Continuo)
26	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Tx] Variable de Control (Calentar)	Control PI (Continuo)
27	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Variable de Control (Enfriar)	Control de 2 puntos
	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Variable de Control (Enfriar)	Control PI (PWM)
28	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Variable de Control (Calentar)	Control de 2 puntos
	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Variable de Control (Calentar)	Control PI (PWM)
29	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Frío Adicional	Temp >= (Consigna+Banda)=> "1"
30	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Calor Adicional	Temp <= (Consigna-Banda)=> "1"
31	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Estado de PI (Enfriar)	0 = Señal PI a 0%; 1 = Señal PI mayor que 0%
32	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Estado de PI (Calentar)	0 = Señal PI a 0%; 1 = Señal PI mayor que 0%
243 - 274	1 Bit	I	C - - W -	DPT_Bool	0/1	[FL] (1 bit) Dato de entrada x	Binary Dato de entrada (0/1)
275 - 290	1 Byte	I	C - - W -	DPT_Value_1_Ucount	0 - 255	[FL] (1 byte) Dato de entrada x	1 byte Dato de entrada (0-255)
291 - 306	2 Bytes	I	C - - W -	DPST-7-1 DPST-8-1 DPST-9-1	0-65585 -32768 - 32767 -273.00 - 670760.00	[FL] (2 bytes) Dato de entrada x	2 bytes Dato de entrada
307 - 314	4 Bytes	I	C - - W -	DPT_Value_4_Count	-2147483648 2147483647	[FL] (4 bytes) Dato de entrada x	4 bytes Dato de entrada
315 - 324	1 Bit	O	C T R - -	DPT_Bool	0/1	[FL] Función X - Resultado	(1 bit) Booleano
	1 Byte	O	C T R - -	DPT_Value_1_Ucount	0 - 255	[FL] Función X - Resultado	(1 byte) Sin signo
	2 Bytes	O	C T R - -	DPT_Value_2_Ucount	0 - 65535	[FL] Función X - Resultado	(2 bytes) Sin signo
	4 Bytes	O	C T R - -	DPT_Value_4_Count	-2147483648 2147483647	[FL] Función X - Resultado	(4 bytes) Con signo
	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[FL] Función X - Resultado	(1 byte) Porcentaje
	2 Bytes	O	C T R - -	DPT_Value_2_Count	-32768 - 32767	[FL] Función X - Resultado	(2 bytes) Con signo
	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[FL] Función X - Resultado	(2 bytes) Flotante

325	1 Bit	O	CTR--	DPT_Bool	0/1	[CC] Todas las válvulas cerradas	0 = Falso; 1 = Verdadero
	1 Bit	O	CTR--	DPT_Bool	0/1	[CC] Todas las válvulas cerradas	0 = Verdadero; 1 = Falso
326	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[CC] Máx. valor de control (salida)	0 - 100 %
327	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[CC] Máx. valor de control (entrada)	0 - 100 %
328, 340, 352, 364, 376, 388, 400, 412	1 Bit	O	CTR--	DPT_Alarm	0/1	[CCx] Error de cortocircuito	0 = No error; 1 = Error
329	1 Bit	O	CTR--	DPT_Alarm	0/1	[CC1-4] Error de sobrecarga	0 = No error; 1 = Error
330, 342, 354, 366, 378, 390, 402, 414	1 Bit	I	C--W-	DPT_Enable	0/1	[CC1] Bloqueo	0 = Desbloquear; 1 = Bloquear
331, 343, 355, 367, 379, 391, 403, 415	1 Bit	I	C--W-	DPT_Alarm	0/1	[CC1] Alarma	0 = No alarma; 1 = Alarma
	1 Bit	I	C--W-	DPT_Alarm	0/1	[CC1] Alarma	0 = Alarma; 1 = No alarma
332, 344, 356, 368, 380, 392, 404, 416	1 Bit	I	C--W-	DPT_Alarm	0/1	[CC1] Alarma 2	0 = No alarma; 1 = Alarma
	1 Bit	I	C--W-	DPT_Alarm	0/1	[CC1] Alarma 2	0 = Alarma; 1 = No alarma
333, 345, 357, 369, 381, 393, 405, 417	1 Bit	I	C--W-	DPT_Trigger	0/1	[CC1] Desenclavar alarma	Alarma = No alarma + Desenclavar (1) -> Fin de alarma
	1 Bit	I	C--W-	DPT_Trigger	0/1	[CC1] Desenclavar alarma	Alarma = Alarma 2 = No alarma + Desenclavar (1) -> Fin de alarma
334, 346, 358, 370, 382, 394, 406, 418	1 Bit	O	CTR--	DPT_Bool	0/1	[CC1] Valor de control - Error	0 = No error; 1 = Error
335, 347, 359, 371, 383, 395, 407, 419	1 Bit	O	CTR--	DPT_State	0/1	[CC1] Protección antiagarre	0 = Inactivo; 1 = Activo
336, 348, 360, 372, 384, 396, 408, 420	1 Bit	I	C--W-	DPT_Switch	0/1	[CC1] Valor de control - 1 bit	0 = Off; 1 = On
337, 349, 361, 373, 385, 397, 409, 421	1 Bit	O	CTR--	DPT_Switch	0/1	[CC1] Valor de control - 1 bit (estado)	0 = Off; 1 = On
338, 350, 362, 374, 386, 398, 410, 422	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[CC1] Valor de control - 1 byte	0 - 100 %
339, 351, 363, 375, 387, 399, 411, 423	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[CC1] Valor de control - 1 byte (estado)	0 - 100 %
377	1 Bit	O	CTR--	DPT_Alarm	0/1	[CC5-8] Error de sobrecarga	0 = No error; 1 = Error

Join and send us your inquiries  
about Zennio devices:

<http://support.zennio.com>

**Zennio Avance y Tecnología S.L.**  
C/ Río Jarama, 132. Nave P-8.11  
45007 Toledo (Spain).

*Tel. +34 925 232 002*

*www.zennio.com*  
*info@zennio.com*



RoHS