



Heating Control

Electromechanical valve controller module

User Manual Version: [1.1]_a

www.zennio.com

CONTENTS

- Contents 2
- 1 Introduction 3
- 2 Configuration..... 4
 - 2.1 Heating Output n..... 8
 - 2.1.1 Configuration..... 8
 - 2.1.2 Alarms..... 13
 - 2.1.3 Cyclical Monitoring..... 15
 - 2.1.4 Startup..... 17
 - 2.1.5 Characteristic Curve 18

1 INTRODUCTION

Some Zennio devices incorporate **heating outputs** that allow independent control of electromechanical valves.

Please refer to the specific user manual and datasheet of each Zennio device in order to confirm whether this feature is available or not, and for specific documentation on the connection and installation of the output loads.

2 CONFIGURATION

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 2.1.1) 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**: in case a short circuit or overload is detected on a particular output, the device reports both errors independently. If a short circuit is detected at an output, it will be periodically notified through a specific object of that output, while overload error is reported jointly for outputs 1-4, through an object per group.

In the event 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) and the associated LED or LEDs will flash every second. This status will be maintained until a new control order is received.

If 4 errors are detected in a 3 minutes period on any of the channels, the device will lock all regulation for the next 3 minutes and the programming LED will flash together with all the output LEDs. There will also be a status object to notify the activation and deactivation of the lock.

- **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

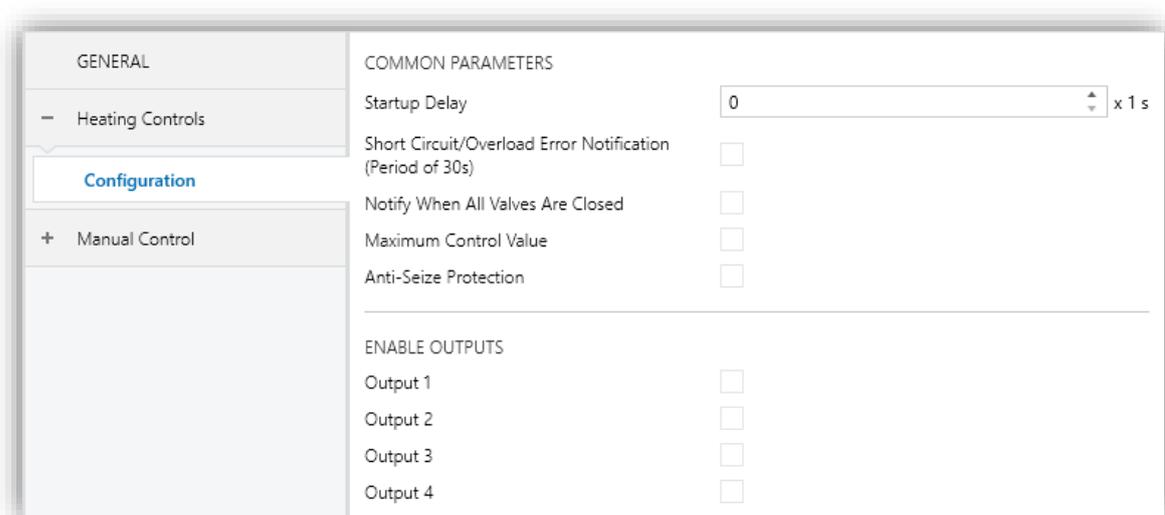


Figure 1. Heating Controls – Configuration.

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

- **Startup Delay** [\[0...255\]\[s\]](#): delay after the device initialisation.
- **Short Circuit/Overload Error Notification** [\[disabled/enabled\]](#): when enabled, the one-bit objects will be available “[HCx] Short Circuit Error” (one for each output) and “[HCx-y] Overload Error” (one for outputs 1-4). 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.

In addition, for both devices, the object “[CC] Short Circuit/Overload Lock”, common to all output, will be enabled. This object will be sent with value “1” when at least 4 error events have been detected in a period of 3 minutes in any of the channels. Three minutes after its activation, the value “0” will be sent, leaving the locking deactivated.

- **Notify When All Valves Are Closed** [*disabled/enabled*]: 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 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 2.1).*

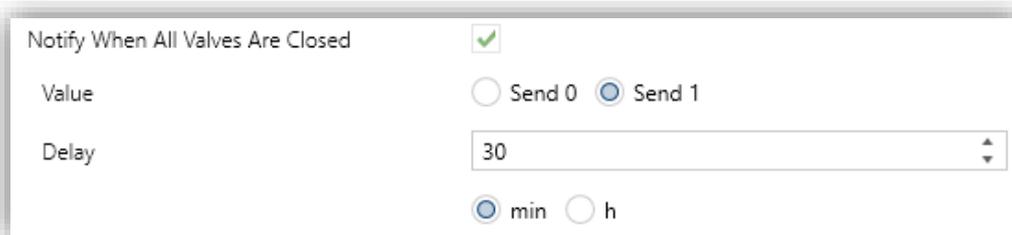


Figure 2. Notify when all valves are closed.

- **Value** [*Send 0 / Send 1*].
- **Delay** [*0...30...1440*][*min*] [*0...24*][*h*].

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 2.1.*

- **Maximum Control Value** [*disabled/enabled*]: 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 2.1).

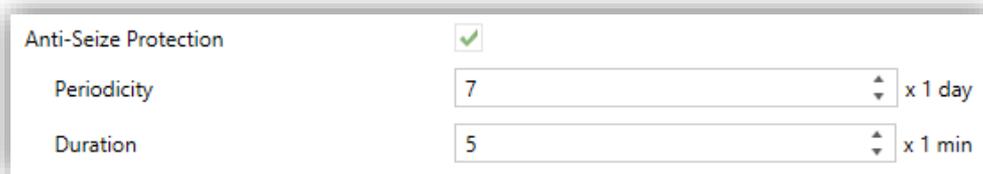


Figure 3. Maximum control value.

- **Include External Value** [*disabled/enabled*]: 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...30...255*][*min*].

Note: *this function ignores those outputs that have not been specifically **marked** for this purpose (see section 2.1).*

- **Anti-Seize Protection** [[disabled/enabled](#)]: 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 x 1 day
Duration	5 x 1 min

Figure 4. Heating Controls – Anti-Seize Protection.

- **Periodicity** [[1...7...30](#)][*day*]: sets the maximum time an output can remain still.
 - **Duration** [[1...5...255](#)][*min*]: sets for how much time the output should switch its position, in case it has been still for more than the above periodicity.
- **Output n** [[disabled/enabled](#)]: 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 2.1.

2.1 HEATING OUTPUT N

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

2.1.1 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 or closing).*

In order to avoid pressure problems in the valves caused by negligible regulations, if the selected control method is percentage, a minimum opening value must be set. Thus, any output command between and the parameterized value, will follow the following logic:

- If the valve is open, the output value will be parameterized.
 - If the valve is closed, it will remain closed.
- **Value for Open Value:** sets which value will command the valve to open.
 - **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 parentage 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	Value for Open Valve	Control order		Output power	LED	Valve state
		1 bit	1 byte			
Normally closed	1	1	100%	Yes		Open
		0	0%	No		Closed
	0	0	100%	Yes		Open
		1	0%	No		Closed
Normally open	1	1	100%	No		Open
		0	0%	Yes		Closed
	0	0	100%	No		Open
		1	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** or to calculate the **maximum control value** (see section 2).
- 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 2.1.2.
- **Cyclical monitoring**: see section 2.1.3.
- **Start-up type**: see section 2.1.4.
- **Characteristic curve**: see section 2.1.5.

ETS PARAMETERISATION

The “Configuration” screen of each individual heating output contains the following parameters:

Figure 5. 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 parameters as well:
 - **PWM Period** [\[5...3600\]\[s\]](#) [\[1...15...1440\]\[min\]](#): sets the PWM cycle time, i.e., the base time for the PWM modulation.

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.

- **Opening Minimum Value** [0...100][%]: minimum value from which the output will start to regulate.

Note: *Even if the control orders come from an alarm, lock, etc., the minimum value criterion will also be applied, so it is recommended not to set fixed regulation values below the minimum in these functionalities.*

Example: *assuming an opening minimum value of 20%, the final status of the valve at the following control orders will be as follows:*

- Valve = 0% → control order = 10% → Valve = 0%.
- Valve = 30% → control order = 10% → Valve = 20%.
- Valve = 0% → alarm with action = 15% → Valve = 0%.
- Valve = 50% → alarm with action = 15% → Valve = 20%

- **Value for Open Valve** [1 / 0]: sets the control value to open the valve. The polarity will also affect the status object of both control methods (1 bit and 1 byte).
- **Valve type** [Normally Closed / Normally Open].

Note: *the control value to close a valve is “0” or “1” (under a one-bit control), as the previous parameter is configured, or always “0%” (under a 1-byte control) no matter if the valve is normally open or normally closed. See Table 1 for details.*

- **Count with this output to:**
 - **Notify That All Valves Are Closed** [disabled/enabled]: 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).*
 - **Calculate the Max. Control Value** [disabled/enabled]: 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).*

- **Show Status Objects** [[disabled/enabled](#)]: 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 2.1.5*

- **Lock** [[disabled/enabled](#)]: 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 / Close Valve / Open Valve](#)].
 - **One-byte control** [[No Change / Specific Control Value](#)]. When selecting the latter, the parameter **Control Value** [[0...100](#)] lets introducing the desired percentage value.
- **Alarms** [[disabled/enabled](#)]: enables or disables the Alarm function. See section 2.1.2.
- **Cyclical Monitoring of the Control Value** [[disabled/enabled](#)]: enables or disables the Cyclical Monitoring function. See section 2.1.3.
- **Characteristic Curve** [[disabled/enabled](#)]: enables or disables the Characteristic Curve function. See section 2.1.5.
- **Startup** [[Default / Custom](#)]: sets whether, during the device initialisation, the output should behave normally or perform a custom action. See section 2.1.4.

2.1.2 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 Alarm 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 2.1.1), 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 tick the corresponding checkbox.

Figure 6. Heating Control – Alarms.

- **Trigger** [0 / 1]: sets the value 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** [0...3600][s] [0...1440][min] [0...24][h]: 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.
- **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](#) / [Close Valve](#) / [Open Valve](#)].

- **1 byte:** [[No Change](#) / [Specific Control Value](#)]. When selecting the latter, the parameter **Control Value** [[0...100](#)] shows up to enter the percentage value.
- **Deactivation – Mode** [[Normal](#) / [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](#) / [Close Valve](#) / [Open Valve](#) / [Last Control Value \(Before the Alarm\)](#)].
 - **1 byte:** [[No Change](#) / [Specific Control Value](#) / [Last Control Value \(Before the Alarm\)](#)]. When selecting “[Specific Control Value]”, the parameter **Control Value** [[0...100](#)] shows up to enter the percentage value.

2.1.3 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 2.1.1), the “Cyclical Monitoring” tab will be available, containing the following parameters.

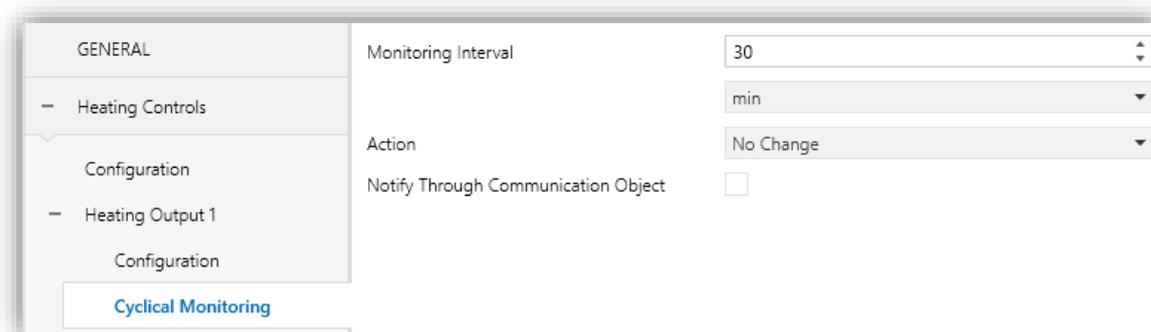


Figure 7. Heating Control – Cyclical Monitoring.

- **Monitoring Interval** [1...3600][s] [1...30...1440][min] [1...24][h].
- **Action**: state the output should acquire if the cyclical monitoring is interrupted. The options depend on the control method:
 - **1 bit**: [No Change / Close Valve / Open Valve].
 - **1 byte**: [No Change / Specific Control Value]. When selecting the latter, the parameter **Control Value** [0...100] shows up to enter the percentage value.
- **Notify Through Communication Object** [disabled/enabled]: 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.

2.1.4 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 2.1.1), a new tab will be incorporated into the tab tree on the left.

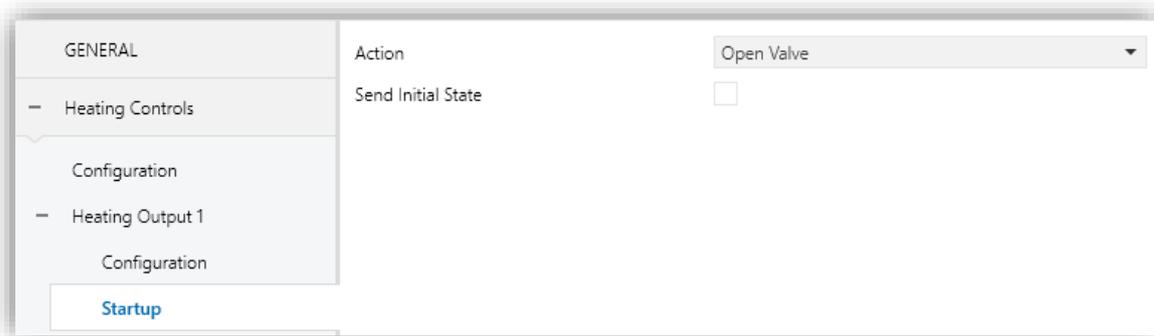


Figure 8 Heating Control - Startup

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 \(Before Bus Failure\)](#) / Close Valve / Open Valve].

➤ **1 byte:** [[Last Control Value \(Before Bus Failure\)](#) / [Specific Control Value](#)].
When selecting the latter, the parameter **Control Value** [0...100] shows up to enter the percentage value.

- **Send Initial State** [[disabled/enabled](#)]: 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...3600][s] [0...1440][min] [0...24][h] to ensure that this value is sent once the other devices are ready to receive it.

Note: This option is only available if **Show Status Object** is enabled from “Configuration” page of each output (see section 2.1.1).

2.1.5 CHARACTERISTIC CURVE

Note: this function is only available under the one-byte control method (see section 2.1.1).

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 9.

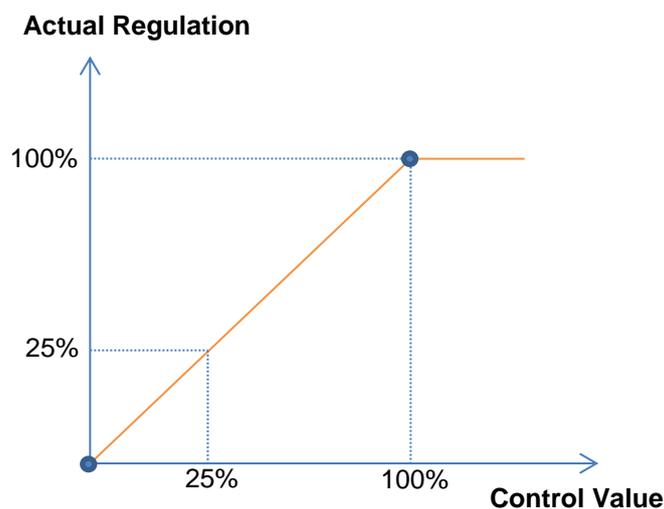


Figure 9. Default Control Curve.

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:

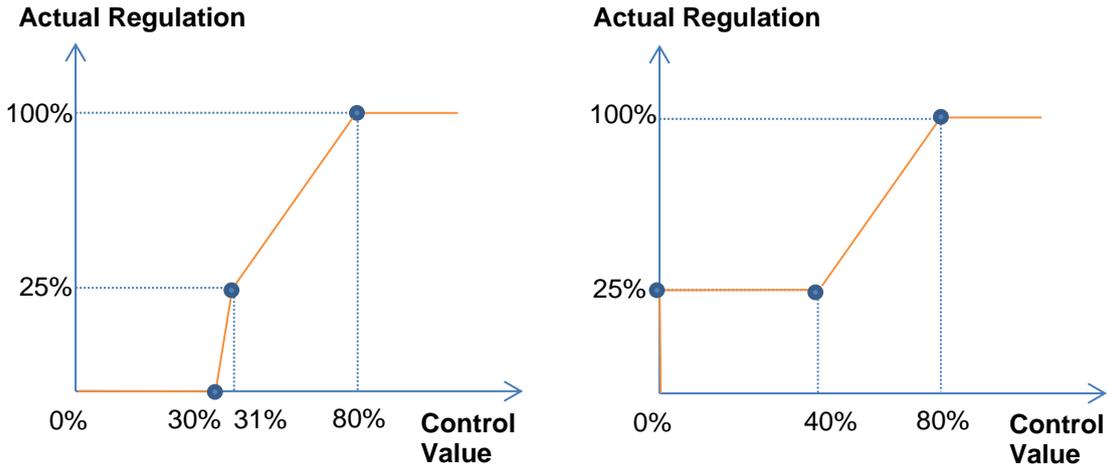


Figure 10. Custom Control Curves.

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 10 and Figure 11):

- For **control values lower than that of the first point**, the actual regulation value will be 0%. For example, in Figure 11 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 11 all control values over 80% imply a regulation value of 100%.

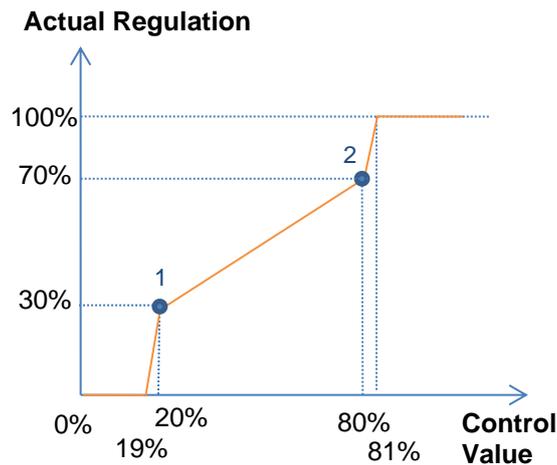


Figure 11. Characteristic Curve – Values Out of Range.

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 2.1.1), a new tab will be incorporated into the tab tree on the left.

Pair	Control Value	Valve Position
Pair 1	0	0
Pair 2	100	100
Pair 3	0	0

⚠ Parametrized values does not correspond with a valid curve. The output may not work properly

Figure 12. Heating Control – Characteristic Curve.

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

- **Control Value** [0...100][%]: input value received through the one-byte control object (value in the “x” axis).
- **Valve Value** [0...100][%]: 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, an error message will appear to indicate the valve behaviour may not be as expected.

Join and send us your inquiries
about Zennio devices:

<https://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