IntesisBox® KNX
MD-AC-KNX-1 v1.0

Interface for integration of Midea air conditioners into KNX TP-1 (EIB) control systems.
Compatible with VRF line air conditioners commercialised by Midea.

Application's Program Version: 1.0

User's Manual
r1 eng

Order Code: **MD-AC-KNX-1**
Issue Date: 2013/06

Intesis Software S.L.
Milà I Fontanals, 1 bis
08700 Igualada
Spain

INDEX

URL http://www.intesis.com
Email info@intesis.com
tel +34 938047134

1 Presentation

MD-AC-KNX-1 allows a complete and natural integration of Midea air conditioners with KNX control systems.

Compatible with all models of VRF line of Midea air conditioners.

Main features:

- Reduced dimensions. Installation even inside the A.C. indoor unit.
- Quick and non visible installation.
- External power not required.
- Direct connection to the KNX EIB bus.
- Direct connection to the AC indoor unit.
- Fully KNX interoperable, configuration from ETS.
- Multiple objects for control (of different types: bit, byte, characters…).
- Special Modes available (Power, Economy, Additional Heating and Additional Cooling).
- Timeout for Open Window and Occupancy. Sleep function also available.
- Control of the AC unit based in the ambient temperature read by the own AC unit, or in the ambient temperature read by any KNX thermostat.
- Total Control and Monitoring of the AC unit from KNX, including monitoring of AC unit's state of internal variables, running hours counter (for filter maintenance control), and error indication and error code.
- AC unit can be controlled simultaneously by the IR remote control of the AC unit and by KNX.
- Up to 5 scenes can be saved and executed from KNX, fixing the desired combination of Operation Mode, Set Temperature, Fan Speed, Vane Position and Remote Controller Lock in any moment by using a simple switching.

2 Connection

The interface comes with a cable (1,9 meters long) for direct connection to the internal control board of the AC indoor unit.

Connection of the interface to the AC indoor unit:

Disconnect mains power from the AC unit. Open the front cover of the indoor unit in order to have access to the internal control board. In the control board locate the socket connector marked as CN9 and remove the connector from the socket. In the control board too, locate the socket connector marked as CN20 and remove the jumper placed in the socket connector. Please keep the jumper in safe place for future use.

Using the cable that comes with the interface, insert its smaller connector into the socket of marked as **CN9**, and the other connector, the biggest one, into the socket marked as **CN20** of the AC unit's control board. Connect the other end of the cable into the MD-AC-KNX-1 socket marked as **AC Unit**.

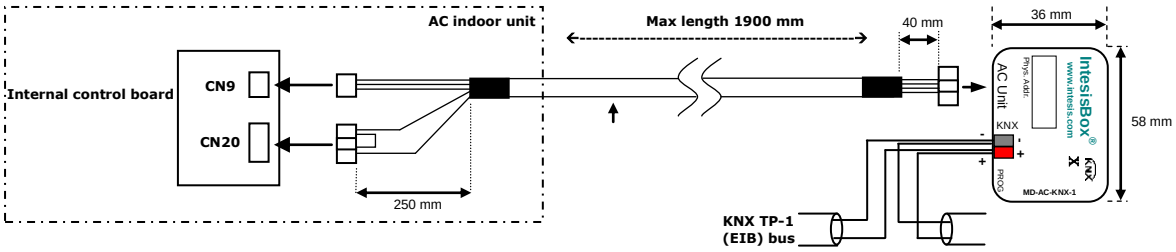
Fix the MD-AC-KNX-1 inside or outside the AC indoor unit depending on your needs – remember that MD-AC-KNX-1 must be also connected to the KNX bus. Close the AC indoor unit's front cover again.

IMPORTANT: Do not modify the length of the cable supplied with the interface, it may affect to the correct operation of the interface.

Connection of the interface to the KNX bus:

Disconnect power of the KNX bus. Connect the interface to the KNX TP-1 (EIB) bus using the KNX standard connector (red/grey) of the interface, respect polarity. Reconnect power of the KNX bus.

Connections diagram:



3 Configuration and setup

This is a fully compatible KNX device which must be configured and setup using standard KNX tool ETS.

ETS project for this device can be donwloaded from:
http://www.intesis.com/down/eib/MD-AC-KNX-1.zip

Please consult the README.txt file, located inside the downloaded zip file, to find instructions on how to install the database.

4 ETS Parameters

When imported to the ETS software for the first time, the gateway shows the following default parameter configuration:

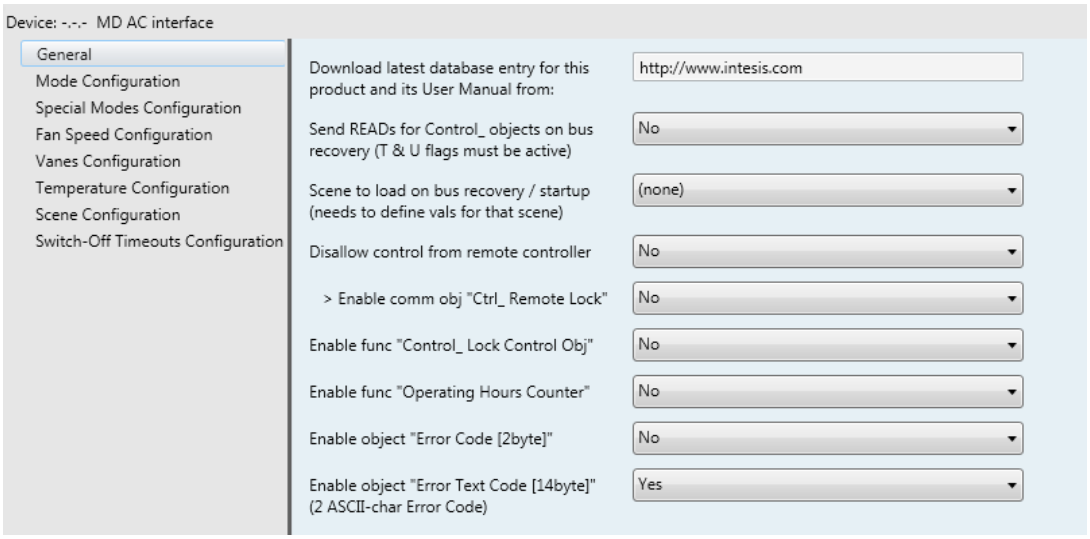


Figure 4.1 Default parameter configuration

With this configuration it's possible to send On/Off (*Control_ On/Off*), change the AC Mode (*Control_ Mode*), the Fan Speed (*Control_ Fan Speed*) and also the Setpoint Temperature (*Control_ Setpoint Temperature*). The Status_ objects, for the mentioned Control_ objects, are also available to use if needed. Also objects *Status_ AC Return Temp* and *Status_ Error/Alarm* are shown.

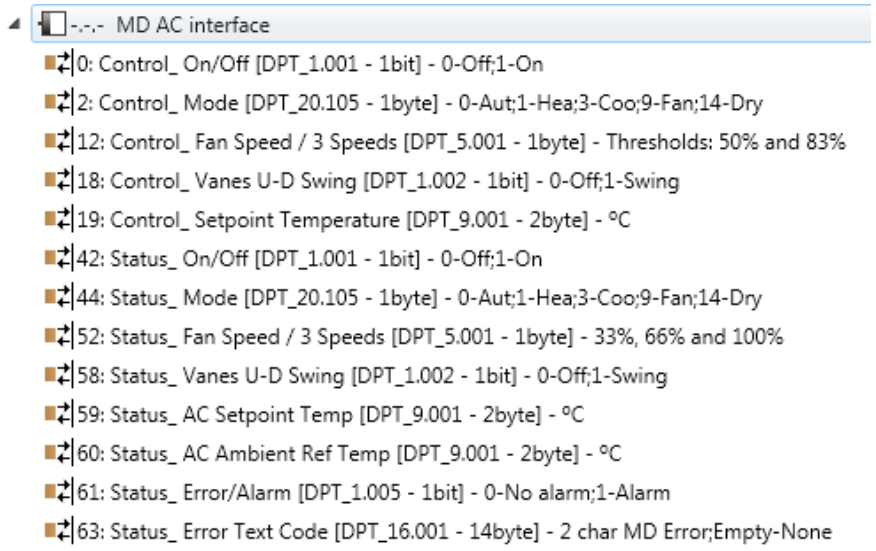


Figure 4.2 Default communication objects

4.1 General dialog

Inside this parameter's dialog it is possible to activate or change the parameters shown in the **Figure 4.1**.

4.1.1 Download latest database entry for this product and its User Manual from:

The first field shows the URL where to download the database and the user manual for the product.

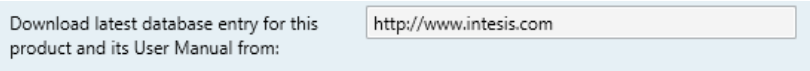


Figure 4.3 Parameter detail

4.1.2 Send READs for Control_ objects on bus recovery

When this parameter is enabled, MD-AC-KNX-1 will send READ telegrams for the group addresses associated on its *Control_* objects on bus recovery or application reset/start-up.

- If set to **"no"** the gateway will not perform any action.
- If set to **"yes"** all *Control_* objects with both Transmit **(T)** and Update **(U)** flags enabled will send READs and their values will be updated with the response when received.

Figure 4.4 Parameter detail

- Delay before sending READs (sec):

  With this parameter, a delay can be configured between 0 and 30 seconds for the READs sent by the *Control_* objects. This is to give time enough to other KNX devices on the bus to start-up before sending the READs.

4.1.3 Scene to load on bus recovery / startup

This parameter executes a selected scene on bus recovery or startup, only if the selected scene has an enabled preset or values previously saved from KNX bus (see Scene Configuration dialog).

If the gateway is disconnected from the indoor unit the scene will not be applied, even when connecting to the indoor unit again.

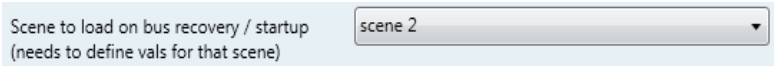


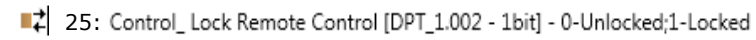
Figure 4.5 Parameter detail

4.1.4 Disallow control from remote controller

This parameter allows:

1- Having the remote controller always locked, or
2- Decide through a new communication object if the RC is locked or not.

- If set to **"yes"** all the actions performed through the remote controller will be disabled.
- If set to **"no"** the remote controller will work as usually. It also appears a new parameter and the communication object *Control_ Lock Remote Control*.



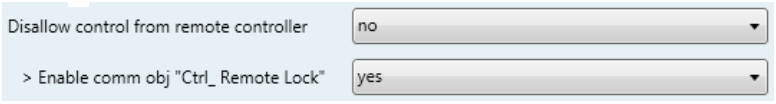


Figure 4.6 Communication object and parameter detail

➢ Enable comm obj "Ctrl_ Remote Lock":

If set to **"no"** the object will not be shown.

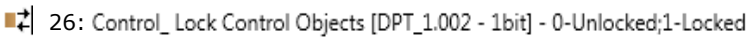
If set to **"yes"** the *Control_ Lock Remote Control* object will appear.

- When a **"1"** value is sent to this communication object, the remote controller is locked. To be unlocked a **"0"** value must be sent. The gateway remembers the last value received even if a KNX bus reset/failure happens.

Important: If an initial scene is enabled and it has as Value for Remote Lock (unchanged) or unlocked, this would unlock the remote controller because the initial scene has priority over the Control_ Lock Remotre Control communication object.

4.1.5 Enable func "Control_ Lock Control Obj"

This parameter shows/hide the *Control_ Lock Control Obj* communication object which, depending on the sent value, locks or unlocks ALL the *Control_* communication objects except itself.

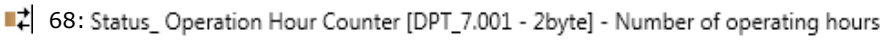


- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Control_ Lock Control Objects* object will appear.

When a **"1"** value is sent to this communication object, all the *Control_* objects will be locked. To unlock a **"0"** value must be sent, as the gateway remembers the last value received even if a KNX bus reset/failure happens.

4.1.6 Enable func "Operating Hours Counter"

This parameter shows/hides the *Status_ Operation Hour Counter* communication object which counts the number of operating hours for the MD-AC-KNX-1.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Status_ Operation Hour Counter* object will appear.
    - This object can be read and sends its status every time an hour is counted. The gateway keeps that count in memory and the status is sent also after a KNX bus reset/failure. Although this object is marked as a *Status_* object it also can be written to update the counter when needed. To reset the counter should be written a **"0"** value.

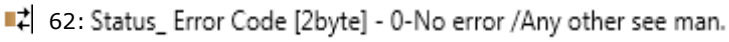
    ⚠ ***Important:*** *This object comes by default without the write **(W)** flag activated. If is necessary to write on it, this flag must be activated.*

    ⚠ ***Important:*** *This object will also return its status, every time a value is written, only if it's different from the existing one.*

    ⚠ ***Important:*** *If the stored value is 0 hours, the gateway will not send the status to KNX.*

4.1.7 Enable object "Error Code [2byte]"

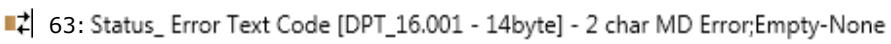
This parameter shows/hides the *Status_ Error Code* communication object which shows the indoor unit errors, if occurred, in numeric format.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Status_ Error Code [2byte]* object will appear.
    - This object can be read and also sends the indoor unit error, if occurred, in numeric format. If a **"0"** value is shown that means no error.

4.1.8 Enable object "Error Text Code [14byte]"

This parameter shows/hides the *Status_ Error Text Code* communication object which shows the indoor unit errors, if occurred, in text format.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Status_ Error Text Code* object will appear.

- This object can be read and also sends the indoor unit error, if occurred, in text format. The errors shown have the same format as in the remote controller and in the error list from the indoor unit manufacturer. If the object's value is empty that means there is no error.

4.2 Mode Configuration dialog

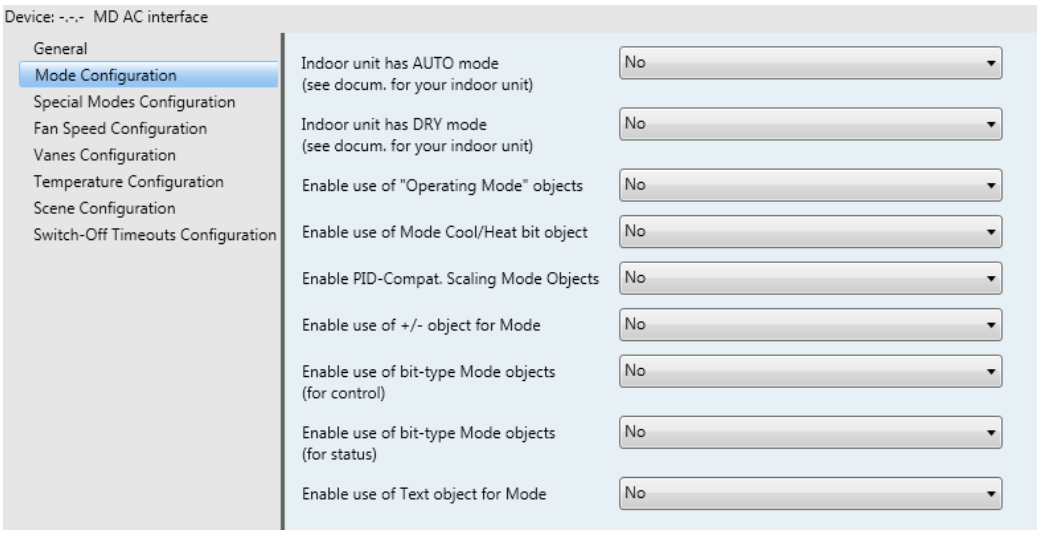
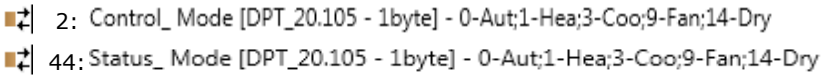


Figure 4.7 Default Mode Configuration dialog

All the parameters in this section are related with the different mode properties and communication objects.



The byte-type communication object for Mode works with the DTP_20.105. Auto mode will be enabled with a **"0"** value, Heat mode with a **"1"** value, Cool mode with a **"3"** value, Fan mode with a **"9"** value and Dry mode with a **"14"** value.

4.2.1 Indoor unit has AUTO mode

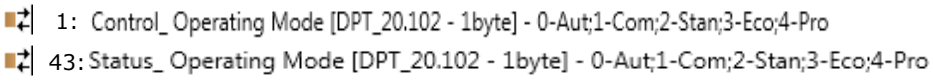
This parameter needs to be set as Yes if the indoor unit has AUTO mode. Please check your indoor unit manual to see wether your unit has or not this AUTO mode.

4.2.2 Indoor unit has DRY mode

This parameter needs to be set as Yes if the indoor unit has DRY mode. Please check your indoor unit manual to see wether your unit has or not this DRY mode.

4.2.3 Enable use of "Operating Mode" objects

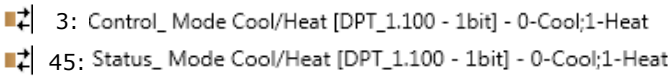
This parameter shows/hides the *Control_* and *Status_ Mode Operating Mode* communication objects.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Control_* and *Status_ Mode Cool/Heat* objects will appear.
    - When a **"1"** value is sent to the *Control_* communication object, **Heat mode** will be enabled in the indoor unit, and the *Status_* object will return this value.
    - When a **"0"** value is sent to the *Control_* communication object, **Cool mode** will be enabled in the indoor unit, and the *Status_* object will return this value.

4.2.4 Enable use of Mode Heat / Cool bit obj

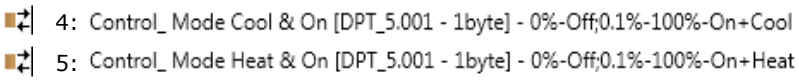
This parameter shows/hides the *Control_* and *Status_ Mode Cool/Heat* communication objects.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Control_* and *Status_ Mode Cool/Heat* objects will appear.
    - When a **"1"** value is sent to the *Control_* communication object, **Heat mode** will be enabled in the indoor unit, and the *Status_* object will return this value.
    - When a **"0"** value is sent to the *Control_* communication object, **Cool mode** will be enabled in the indoor unit, and the *Status_* object will return this value.

4.2.5 Enable PID-Compat. Scaling Mode Objects

This parameter shows/hides the *Control_ Mode Cool & On* and *Control_ Mode Heat & On* communication objects.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Control_ Mode Cool & On* and *Control_ Mode Heat & On* objects will appear.
    - These objects provide compatibility with those KNX thermostats that control the demand of heating or cooling by using scaling (percentage) objects. In these thermostats, the percentage demand is meant to be applied on a fluid valve of the heating / cooling system.

- MD-AC-KNX-1 device does not provide individual control on the internal parts of the indoor unit (as can be its compressor, refrigerant valves, etc). Rather, it provides the same level of control as a (user) remote controller.

- Objects "Control_ Mode Cool & On" and "Control_ Mode Heat & On" intend to bring compatibility between thermostats oriented to the control of custom heating / cooling systems and ready-made AC indoor units, by applying the following logic:

  - Whenever a non-zero value (>0%) is received at "Control_ Mode Cool & On", indoor unit will switch On in COOL mode.

  - Whenever a non-zero value (>0%) is received at "Control_ Mode Heat & On", indoor unit will switch On in HEAT mode.

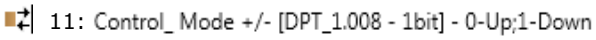
    - Lastest updated object will define the operating mode

  - Indoor unit will switch off only when both objects become zero (0%) – or when an OFF is requested at object "0. On/Off [DPT_1.001 - 1bit]"

⚠ ***Important:*** *These objects function is only to send On/Off and Cool/Heat to the indoor unit. The PID (Inverter system) is calculated by the indoor unit itself. Please consider introducing an appropriate PID configuration to the external KNX thermostat to not interfere the indoor unit PID.*

4.2.6 Enable use of + / - object for Mode

This parameter shows/hides the *Control_ Mode +/-* communication object which lets change the indoor unit mode by using two different datapoint types.



- If set to **"no"** the object will not be shown.

- If set to **"yes"** the *Control_ Mode +/-* object and a new parameter will appear.

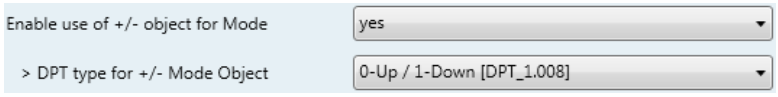


Figure 4.8 Parameter detail

➢ DPT type for +/- Mode Object

This parameter lets choose between the datapoints **0-Up / 1-Down [DPT_1.008]** and **0-Decrease / 1-Increase [DPT_1.007]** for the *Control_ Mode +/-* object.

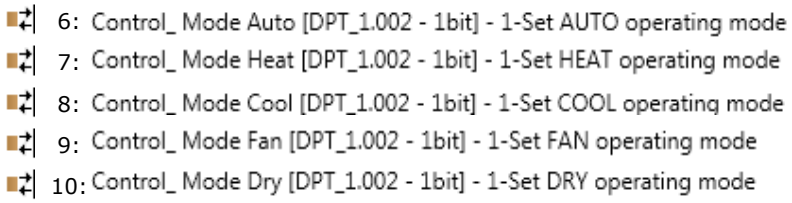
The sequence followed when using this object is shown below:

- Up / Increase
- Down / Decrease

Keep in mind that depending on the indoor unit you have and the available features, Auto mode and Dry mode my not be present. See section 4.2.1 and 4.2.2 for more details.

4.2.7 Enable use of bit-type Mode objects (for control)

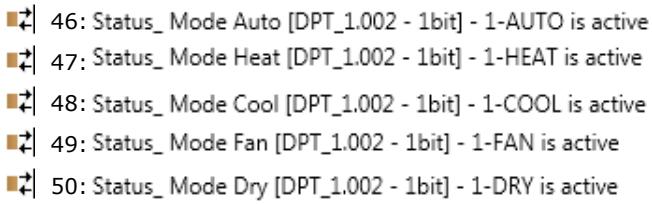
This parameter shows/hides the bit-type *Control_ Mode* objects.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Control_ Mode* objects for Auto, Heat, Cool, Fan and Dry will appear. To activate a mode by using these objects a **"1"** value has to be sent.

4.2.8 Enable use of bit-type Mode objects (for status)

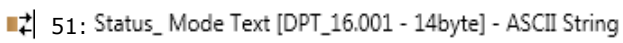
This parameter shows/hides the bit-type *Status_ Mode* objects.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Status_ Mode* objects for Auto, Heat, Cool, Fan and Dry will appear. When enabled, a mode will return a **"1"** through its bit-type object.

4.2.9 Enable use of Text object for Mode

This parameter shows/hides the *Status_ Mode Text* communication object.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Status_ Mode Text* object will appear. Also, in the parameters, will be shown five text fields, one for each mode, that will let modify the text string displayed by the *Status_ Mode Text* when changing mode.

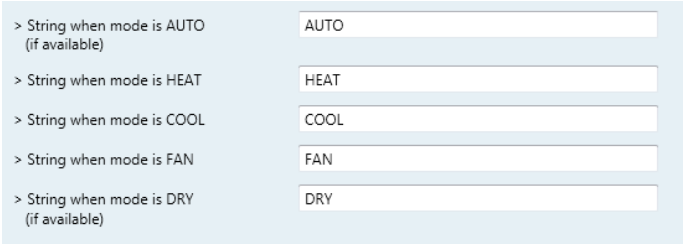


Figure 4.9 Parameter detail

4.3 Special Modes Configuration dialog

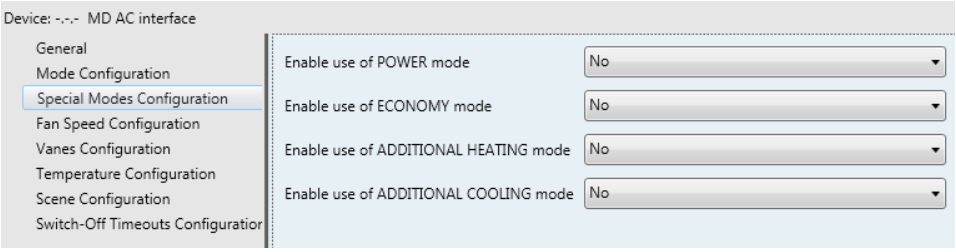


Figure 4.10 Parameter detail

The Special Modes can be parameterized through the ETS parameters dialog, and they can be used to give extra functionality.

⚠ ***Important:*** *When executing any of the Special Modes, the real state of the indoor unit will NOT be shown in KNX.*

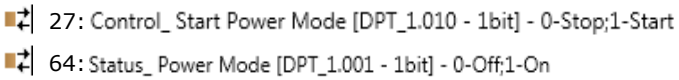
⚠ ***Important:*** *When the predefined time for the Special Mode is finished or a **"0"** value is sent to stop it; the previous state will be recovered.*

⚠ ***Important:*** *If a value concerning On/Off, Mode, Fan Speed or Setpoint Temperature is received from KNX while any Special Mode is running (**"1"**), the Special Mode will stop and the previous state will be recovered. The value received will be also applied then.*

⚠ ***Important:*** *If a value concerning On/Off, Mode, Fan Speed or Setpoint Temperature is modified through the remote controller, the Special Mode will stop WITHOUT recovering the previous state. Then the real indoor unit state will be shown in KNX including the new value received through the remote controller.*

4.3.1 Enable use of POWER mode

This parameter shows/hides the *Control_ Start Power Mode* and *Status_ Power Mode* communication objects. The Power Mode lets change the Setpoint Temperature and the Fan Speed within a given period of time.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Control_ Start Power Mode* and *Status_ Power Mode* objects and new parameters will appear.

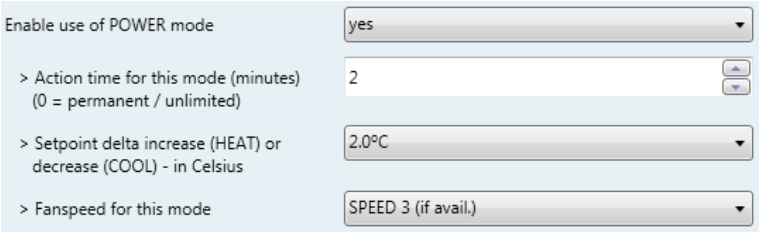


Figure 4.11 Parameter detail

- When a **"1"** value is sent to the *Control_* communication object Power Mode will be enabled, and the *Status_* object will return this value.
- When a **"0"** value is sent to the *Control_* communication object, Power Mode will be disabled, and the *Status_* object will return this value.

⚠ ***Important:*** *This mode will ONLY work if the indoor unit is both turned on and in a Heat, Cool, Auto-Heat or Auto-Cool Mode.*

➢ Action time for this mode (minutes):

Duration of Power Mode, in minutes, once started.

➢ Setpoint delta increase (HEAT) or decrease (COOL) – in Celsius:

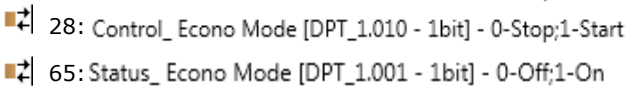
Number of degrees Celsius that will increase in Heat Mode, or decrease in Cool Mode, while in Power Mode.

➢ Fan Speed for this mode:

Fan Speed that will be set in the unit while in Power Mode.

4.3.2 Enable use of ECONOMY mode

This parameter shows/hides the *Control_ Start Econo Mode* and *Status_ Econo Mode* communication objects. The Econo Mode lets change the Setpoint Temperature and the Fan Speed within a given period of time.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Control_ Start Econo Mode* and *Status_ Econo Mode* objects and new parameters will appear.
  - When a **"1"** value is sent to the *Control_* communication object, EconoMode will be enabled, and the *Status_* object will return this value.
  - When a **"0"** value is sent to the *Control_* communication object, EconoMode will be disabled, and the *Status_* object will return this value.

⚠ ***Important:*** *This mode will ONLY work if the indoor unit is both turned on and in a Heat, Cool, Auto-Heat or Auto-Cool Mode.*

- Action time for this mode (minutes):

  Duration of EconoMode, in minutes, once started.

- Setpoint delta increase (HEAT) or decrease (COOL) – in Celsius:

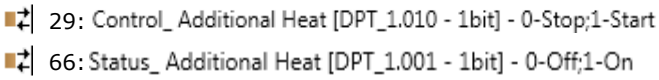
  Number of degrees Celsius that will increase in Heat Mode, or decrease in Cool Mode, while in EconoMode.

- Fan Speed for this mode:

  Fan Speed that will be set in the unit while in EconoMode.

4.3.3 Enable use of ADDITIONAL HEATING mode

This parameter shows/hides the *Control_ Start Additional Heat Mode* and *Status_ Additional Heat Mode* communication objects. The Additional Heating Mode lets change the Setpoint Temperature and the Fan Speed within a given period of time.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the Control_ Start Additional Heat Mode and Status_ Additional Heat Mode objects and new parameters will appear.

  - When a **"1"** value is sent to the *Control_* communication object, Additional Heating Mode will be enabled, and the *Status_* object will return this value.
  - When a **"0"** value is sent to the *Control_* communication object, Additional Heating Mode will be disabled, and the *Status_* object will return this value.

  ⚠ ***Important:*** *This mode will ALWAYS turn on the indoor unit in Heat mode.*

- Action time for this mode (minutes):

  Duration of Additional Heating Mode, in minutes, once started.

- Setpoint temp for this mode (ºC):

  Setpoint temperature that will be applied while in Additional Heating Mode.

- Fan Speed for this mode:

  Fan Speed that will be set in the unit while in Additional Heating Mode.

4.3.4 Enable use of ADDITIONAL COOLING mode

This parameter shows/hides the *Control_ Start Additional Cool Mode* and *Status_ Additional Cool Mode* communication objects. The Additional Heating Mode lets change the Setpoint Temperature and the Fan Speed within a given period of time.

30: Control_ Additional Cool [DPT_1.010 - 1bit] - 0-Stop;1-Start
67: Status_ Additional Cool [DPT_1.001 - 1bit] - 0-Off;1-On

- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the Control_ Start Additional Cool Mode and Status_ Additional Cool Mode objects and new parameters will appear.
  - When a **"1"** value is sent to the *Control_* communication object, Additional Cooling Mode will be enabled, and the *Status_* object will return this value.
  - When a **"0"** value is sent to the *Control_* communication object, Additional Cooling Mode will be disabled, and the *Status_* object will return this value.

  ⚠ ***Important:*** *This mode will ALWAYS turn on the indoor unit in Cool mode.*

➢ Action time for this mode (minutes):

Duration of Additional Cooling Mode, in minutes, once started.

➢ Setpoint temp for this mode (ºC):

Setpoint temperature that will be applied while in Additional Cooling Mode.

➢ Fan Speed for this mode:

Fan Speed that will be set in the unit while in Additional Cooling Mode.

4.4 Fan Speed Configuration dialog

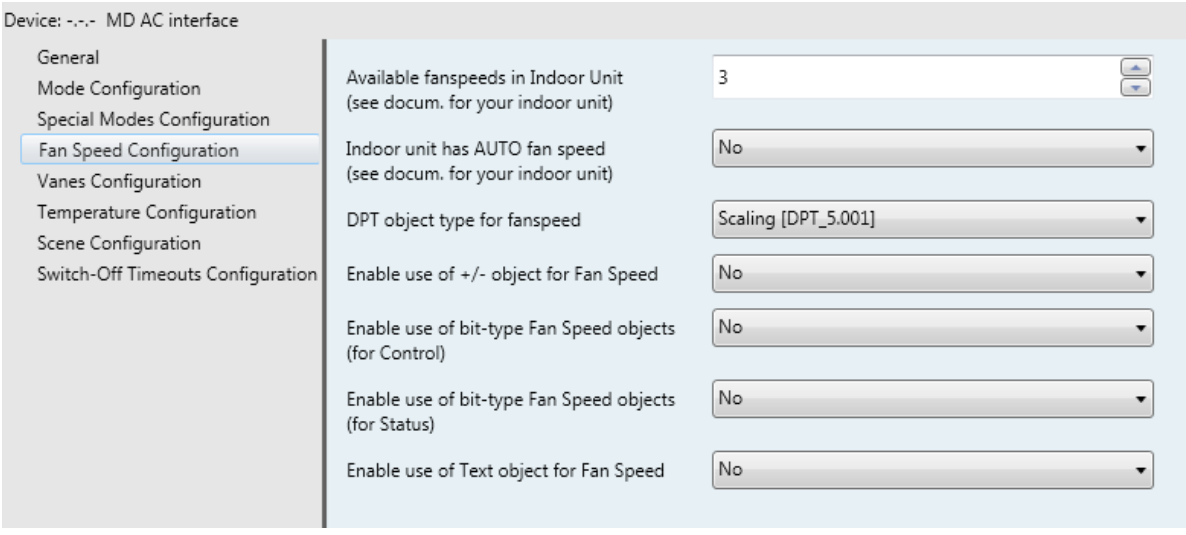


Figure 4.12 Default Fan Speed Configuration dialog

All the parameters in this section are related with the Fan Speed properties and communication objects.

4.4.1 Available fan speeds in Indoor Unit

This parameter lets you choose how many fan speeds are available in the indoor unit (2 or 3 speeds available).

Figure 4.13 Parameter detail

Changing the fan speeds will also change the fan speed byte-type object (and the bit-type objects) erasing all the group addresses associated.

⚠ ***Important:*** *Read the documentation of your indoor unit to check how many fan speeds are available.*

4.4.2 Indoor unit has AUTO fan speed

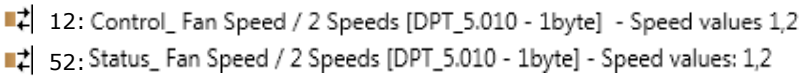
This parameter needs to be set as Yes if the indoor unit has AUTO mode. Please check your indoor unit manual to see wether your unit has or not this AUTO mode.

4.4.3 DPT object type for fanspeed

With this parameter is possible to change de DPT for the *Control_ Fan Speed* and *Status_ Fan Speed* byte-type communication objects. Datapoints Scaling (DPT_5.001) and Enumerated (DPT_5.010) can be selected.

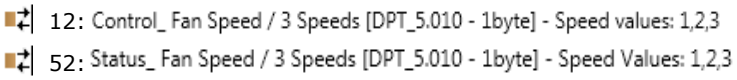
- When **"Enumerated [DPT 5.010]"** is selected, *Control_ Fan Speed* and *Status_ Fan Speed* communication objects for this DPT will appear. Also, depending on the number of fan speeds selected, these objects will be different.

  If this DPT is selected with 2 fan speeds:

  The first fan speed will be selected if a **"1"** is sent to the *Control_* object. The second fan speed will be selected sending a **"2"**.

  The *Status_* object will always return the value for the fan speed selected.
  If this DPT is selected with 3 fan speeds:

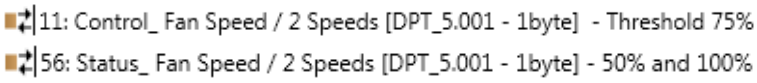
The first fan speed will be selected if a **"1"** is sent to the *Control_* object. The second one will be selected sending a **"2"**, and the last one sending a **"3"**.

The *Status_* object will always return the value for the fan speed selected.

⚠ ***Important:*** *In both cases if a **"0"** value is sent to the Control_ object, the minimum fan speed will be selected. If a value bigger than **"2"** (in case of 2 speeds) or bigger than **"3"** (in case of 3 fan speeds) is sent to the Control_ object, then the maximum fan speed will be selected.*

- When **"Scaling [DPT 5.001]"** is selected, *Control_ Fan Speed* and *Status_ Fan Speed* communication objects for this DPT will appear. Also, depending on the number of fan speeds selected, these objects will be different.

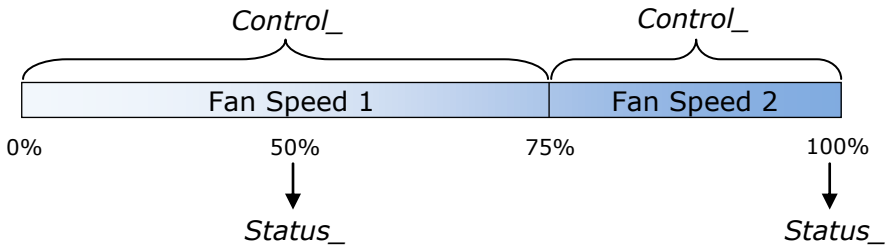
If this DPT is selected with 2 fan speeds:



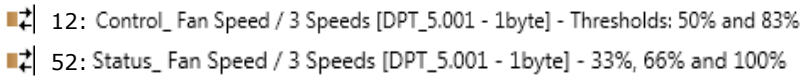
When a value between **0%** and **74%** is sent to the *Control_* object the first fan speed will be selected.

When a value between **75%** and **100%** is sent to the *Control_* object, the second speed will be selected.

The *Status_* object will return a **50%** for the first fan speed, and a **100%** for the second one.



If this DPT is selected with 3 fan speeds:

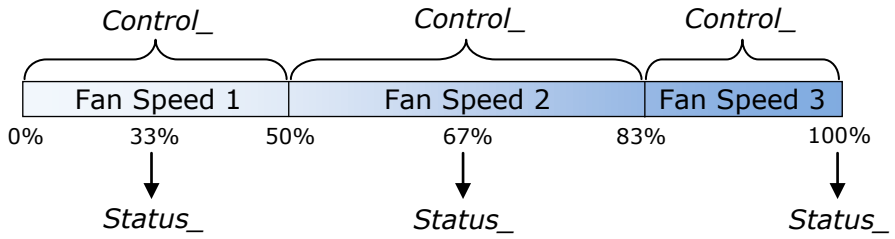


When a value between **0%** and **49%** is sent to the *Control_* object the first fan speed will be selected.

When a value between **50%** and **83%** is sent to the *Control_* object, the second speed will be selected.

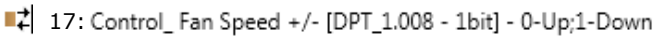
When a value between **84%** and **100%** is sent to the *Control_* object, the third speed will be selected.

The *Status_* object will return a **33%** when the first speed is selected, a **67%** for the second one and a **100%** for the third one.



4.4.4 Enable use of +/- object for Fan Speed

This parameter shows/hides the *Control_ Fan Speed +/-* communication object which lets you increase/decrease the indoor unit fan speed by using two different datapoint types.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Control_ Fan Speed +/-* object and a new parameter will appear.

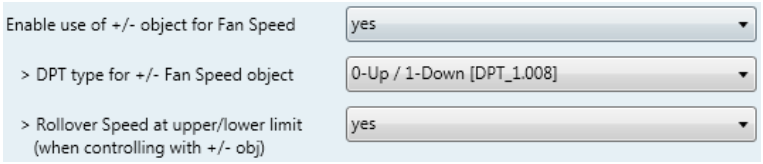


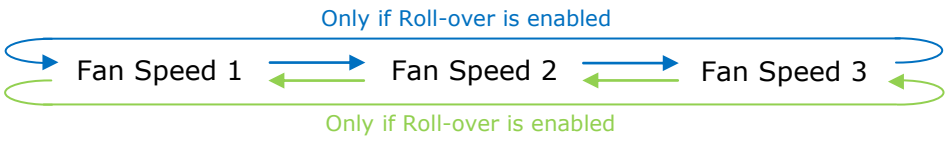
Figure 4.14 Parameter detail

➢ DPT type for +/- Fan Speed Object

This parameter lets choose between the datapoints **0-Up / 1-Down [DPT_1.008]** and **0-Decrease / 1-Increase [DPT_1.007]** for the *Control_ Fan Speed +/-* object.

➢ Roll-over Speed at upper/lower limit

This parameter lets choose if roll-over will be enabled (**"yes"**) or disabled (**"no"**) for the *Control_ Fan Speed +/-* object.

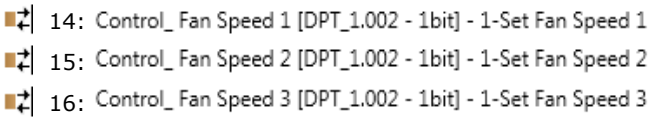


- Up / Increase
- Down / Decrease

URL http://www.intesis.com
Email info@intesis.com
tel +34 938047134

4.4.5 Enable use of bit-type Fan Speed objects (for Control)

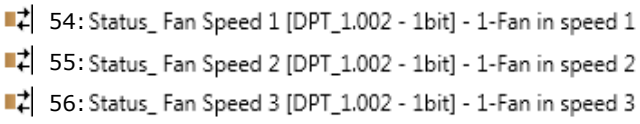
This parameter shows/hides the bit-type *Control_ Fan Speed* objects.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Control_ Fan Speed* objects for Speed 1, Speed 2 and Speed 3 (if available) will appear. To activate a Fan Speed by using these objects a **"1"** value has to be sent.

4.4.6 Enable use of bit-type Fan Speed objects (for Status)

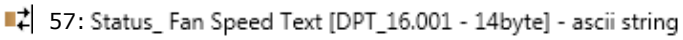
This parameter shows/hides the bit-type *Status_ Fan Speed* objects.



- If set to **"no"** the objects will not be shown.
- If set to **"yes"** the *Status_ Fan Speed* objects for Speed 1, Speed 2 and Speed 3 (if available) will appear. When a Fan Speed is enabled, a **"1"** value is returned through its bit-type object.

4.4.7 Enable use of Text object for Fan Speed

This parameter shows/hides the *Status_ Fan Speed Text* communication object.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Status_ Fan Speed Text* object will appear. Also, in the parameters, will be shown two (or three, depending on the number of fan speeds selected) text fields, one for each Fan Speed, that will let modify the text string displayed by the *Status_ Fan Speed Text* when changing a fan speed.

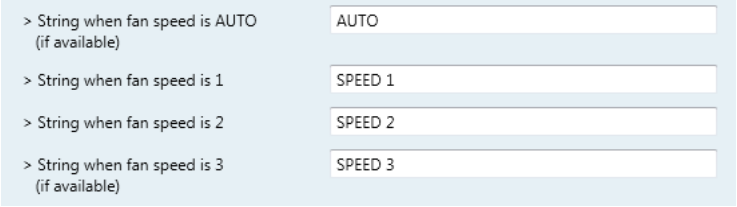


Figure 4.15 Parameter detail

4.5 Vanes Configuration

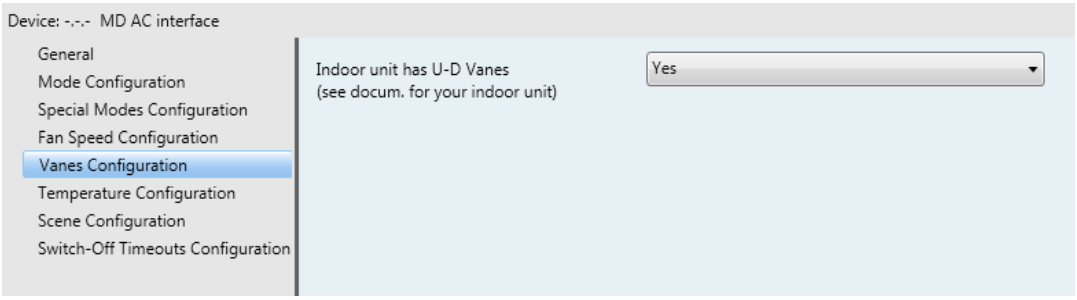


Figure 4.16 Vane Up-Down configuration dialog

All the parameters in this section are related with the Vane Up-Down properties and communication objects.

4.5.1 Indoor unit has U-D Vanes

This parameter lets you choose if the unit has Up-Down Vanes available or not.

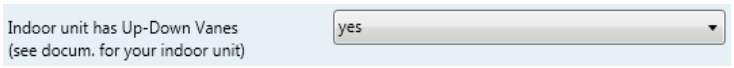
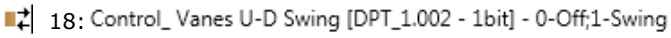


Figure 4.17 Parameter detail

- If set to **"no"** all the parameters and communication objects for the Up-Down Vanes will not be shown.
- If set to **"yes"** all the parameters and communication objects (if enabled in the parameters dialog) for the Up-Down Vanes will be shown.



⚠ ***Important:*** *Read the documentation of your indoor unit to check if Up-Down Vanes are available.*

4.6 Temperature Configuration dialog

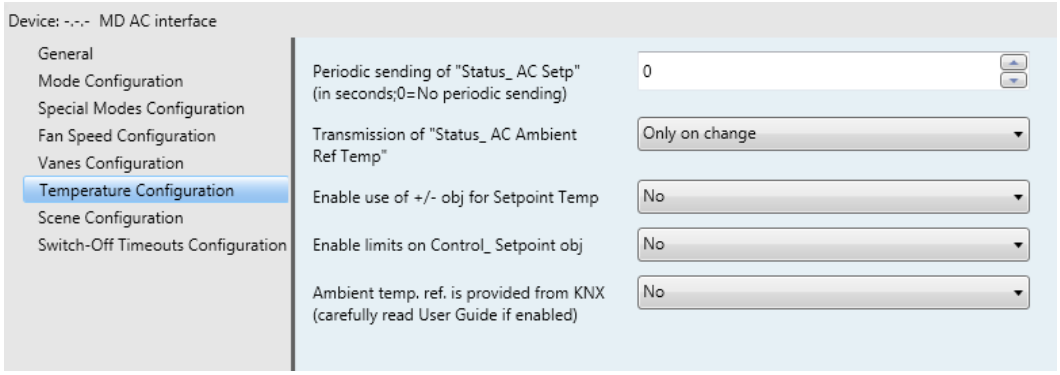


Figure 4.20 Default Temperature Configuration dialog

All the parameters in this section are related with the Temperature properties and communication objects.

4.6.1 Periodic sending of "Status_ AC Setp"

This parameter lets you change the interval of time (in seconds, from 0 to 255) at the end of which the AC setpoint temperature is sent to the KNX bus. For a **"0"** value, the AC setpoint temperature will ONLY be sent on change. The AC setpoint temperature is sent through the communication object *Status_ AC Setpoint Temp.*

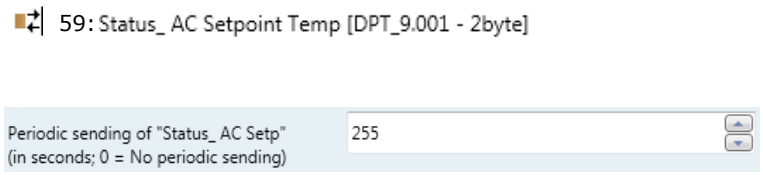


Figure 4.21 Parameter detail

⚠ ***Important:*** *In case of working with the gateway in slave mode and the ambient temperature provided from KNX, the setpoint temperature returned from this object, will be the one resulting from the formula shown in the section "4.6.5 Ambient temp. ref. is provided from KNX".*

4.6.2 Transmission of "Status_ AC Ref Temp"

This parameter lets to you choose if the AC return temperature will be sent **"only cylically"**, **"only on change"** or **"cyclically and on change"**. The AC return temperature is sent through the communication object *Status_ AC Return Temp.*

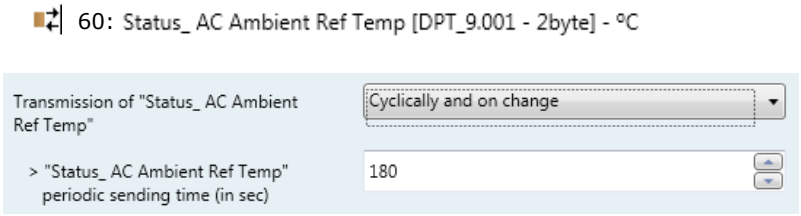


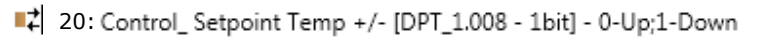
Figure 4.22 Parameter detail

- "Status_ AC SetTemp" periodic sending time (in sec)

  This parameter will only be available for the **"only cyclically"** and **"cyclically and on change"** options, and lets you change the interval of time (in seconds, from 1 to 255) at the end of which the AC return temperature is sent to the KNX bus.

4.6.3 Enable use of +/- obj for Setp Temp

This parameter shows/hides the *Control_ Setpoint Temp +/-* communication object which lets you change the indoor unit setpoint temperature by using two different datapoint types.



- If set to **"no"** the object will not be shown.

- If set to **"yes"** the *Control_ Setpoint Temp +/-* object and a new parameter will appear.

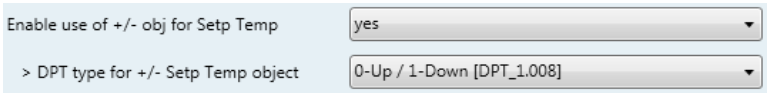


Figure 4.23 Parameter detail

➢ DPT type for +/- Setp Temp object

This parameter lets choose between the datapoints **0-Up / 1-Down [DPT_1.008]** and **0-Decrease / 1-Increase [DPT_1.007]** for the *Control_ Setpoint Temp +/-* object.

(Lower limit) **16ºC** ⇄ 17ºC ⇄ … ⇄ 31ºC ⇄ **32ºC** (Upper limit)

- Up / Increase
- Down / Decrease

4.6.4 Enable limits on Control_ Setpoint obj

This parameter enables to define temperature limits for the *Control_ Setpoint Temperature* object.

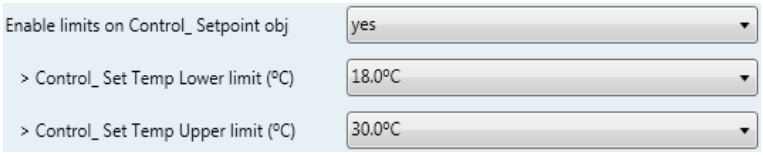


Figure 4.24 Parameter detail

- If set to **"no"** the setpoint temperature limits for the *Control_ Setpoint Temperature* object will be the default: 16ºC for the lower limit and 32ºC for the upper limit.
- If set to **"yes"** it is possible to define temperature limits for the *Control_ Setpoint Temperature* object.

➢ Control_ Set Temp Lower limit (ºC)

This parameter lets to define the lower limit for the setpoint temperature.

➢ Control_ Set Temp Upper limit (ºC)

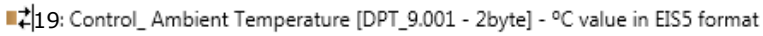
This parameter lets to define the upper limit for the setpoint temperature.

⚠ ***Important:*** *If a setpoint temperature above the upper defined limit (or below the lower defined limit) is sent through the Control_ Setpoint Temperature object, it will be ALWAYS applied the limit defined.*

⚠ ***Important:*** *When limits are enabled, any setpoint temperature sent to the AC (even through scenes, special modes, etc.) will be limited.*

4.6.5 Ambient Ref. Temp. is provided from KNX

This parameter shows/hides the *Control_ Ambient Temperature* communication object which lets you use an ambient temperature reference provided by a KNX device.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Control_ Ambient Temperature* object will appear. Meant to be enabled when you want the temperature provided by a KNX sensor to be the reference ambient temperature for the air conditioner. Then, the following formula applies for the calculation of real *Control_ Setpoint Temperature* sent ot the AC unit:

  "AC Setp. Temp" = "Ambient ref. Temp" - ("KNX Amb. Temp." - "KNX Setp.

  - AC Setp. Temp: AC indoor unit setpoint temperature
  - Ambient Ref. Temp: AC indoor unit return temperature
  - KNX Amb. Temp.: Ambient temperature provided from KNX
  - KNX Setp. Temp: Setpoint temperature provided from KNX

  As an example, consider the following situation:

  User wants: **19ºC** ("KNX Setp. Temp.")
  User sensor (a KNX sensor) reads: **21ºC** ("KNX Amb Temp.")
  Ambient temp. read by Midea system is: **24ºC** ("Ambient Ref. Temp")

  In this example, the final setpoint temperature that MD-AC-KNX-1 will send out to the indoor unit (shown in "Setp. Temp.") will become 24ºC – (21ºC - 19ºC) = **22ºC**. This is the setpoint that will actually be requested to Midea unit.

  This formula will be applied as soon as the *Control_ Setpoint Temperature* and *Control_ Ambient Temperature* objects are written at least once from the KNX installation. After that, they are kept always consistent.

  Note that this formula will always drive the AC indoor unit demand in the *right* direction, regardless of the operation mode (Heat, Cool or Auto).

4.7 Scene Configuration dialog

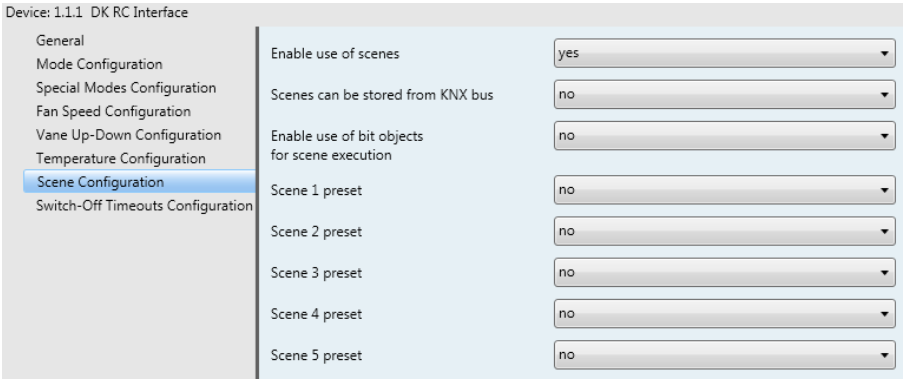


Figure 4.25 Parameter detail

All the parameters in this section are related with the Scene properties and communication objects. A scene contains values of: On/Off, Mode, Fan speed, Vane position, Setpoint Temperature and Remote Controller Disablement.

4.7.1 Enable use of scenes

This parameter shows/hides the scene configuration parameters and communication objects.

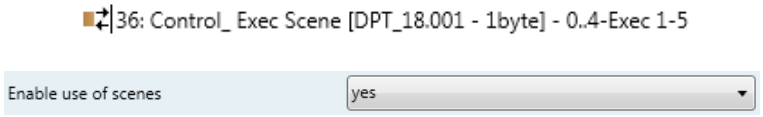


Figure 4.26 Parameter detail

- If set to **"no"** the scene parameters and communication objects will not be shown.
- If set to **"yes"** the scene parameters and communication objects will be shown. To execute a scene through the byte-type object, a value from **"0"** to **"4"** has to be sent, correponding each one to a different scene (i.e. "0" = Scene 1;… "4" = Scene 5).

4.7.2 Scenes can be stored from KNX bus

This parameter shows/hides the *Control_ Save/Exec Scene* and all the *Control_ Store Scene* (if enabled) communication objects.

- If set to **"no"** the communication objects will not be shown.
- If set to **"yes"** the communication objects and a new parameter will appear. To store a scene through the byte-type object, a value from **"128"** to **"132"** has to be sent to the object, correponding each one to a different scene (i.e. "128" = Scene 1;… "132" = Scene 5).

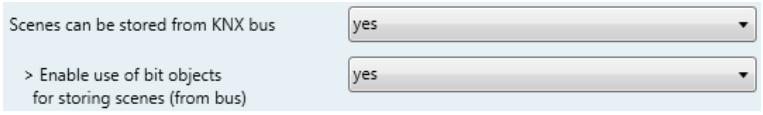


Figure 4.27 Parameter detail

- Enable use of bit objects for storing scenes (from bus)

  If set to **"no"** the objects will not be shown.

If set to **"yes"** the *Control_ Store Scene* objects for storing scenes will appear. To store a scene by using these objects, a **"1"** value has to be sent to the scene's object we want to store (i.e. to store scene 4, a "1" has to be sent to the *Control_ Store Scene 4* object).

4.7.3 Enable use of bit objects for scene execution

This parameter shows/hides the *Control_ Execute Scene* bit-type communication objects.

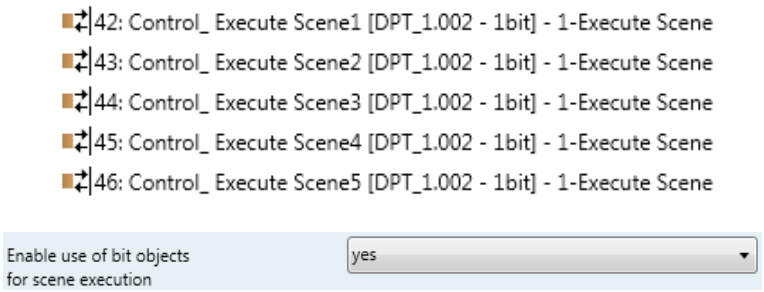


Figure 4.28 Parameter detail

- If set to **"no"** the communication objects will not be shown.
- If set to **"yes"** the communication objects will appear. To execute a scene by using these objects, a **"1"** value has to be sent to the scene's object we want to execute (i.e. to execute scene 4, a "1" has to be sent to the *Control_ Execute Scene 4* object).

4.7.4 Scene "x" preset

This parameter lets you define a preset for a scene (the following description is valid for all the scenes).

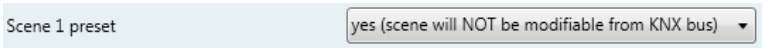


Figure 4.29 Parameter detail

- If set to **"no"** the preset for the scene "x" will be disabled.
- If set to **"yes"** the preset will be enabled. When a scene is executed the values configured in the preset will be aplied.

⚠ ***Important:*** *If a scene's preset is enabled, will not be possible to modify (store) the scene from the KNX bus.*

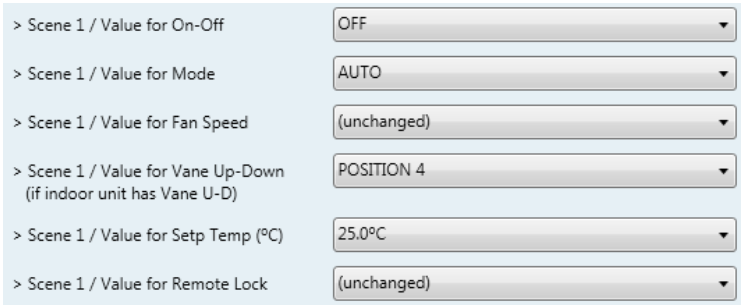


Figure 4.30 Parameter detail

- Scene "x" / Value for On-Off

  This parameter lets you choose the power of the indoor unit when the scene is executed. The following options are available: **"ON", "OFF"** or **"(unchanged)"**.

- Scene "x" / Value for Mode

  This parameter lets you choose the mode of the indoor unit when the scene is executed. The following options are available: **"AUTO"**, **"HEAT"**, **"COOL"**, **"FAN"**, **"DRY"**, or **"(unchanged)"**.

- Scene "x" / Value for Fan Speed

  This parameter lets you choose the fan speed of the indoor unit when the scene is executed. The following options are available: **"SPEED 1"**, **"SPEED 2"**, **"SPEED 3"**, or **"(unchanged)"**.

- Scene "x" / Value for Vane Up-Down

  This parameter lets you choose the vane position of the indoor unit when the scene is executed. The following options are available: **"POSITION 1"**, **"POSITION 2"**, **"POSITION 3"**, **"POSITION 4", "POSITION 5", "SWING"** or **"(unchanged)"**.

- Scene "x" / Value for Setp Temp (ºC)

  This parameter lets you choose the setpoint temperature of the indoor unit when the scene is executed. The following options are available: from **"16ºC"** to **"32ºC"** (both included), or **"(unchanged)"**.

- Scene "x" / Value for Remote Lock

  This parameter lets you choose the remote controller status of the indoor unit when the scene is executed. The following options are available: **"locked"**, **"unlocked"**, or **"(unchanged)"**.

⚠ ***Important:*** *If any preset value is configured as **"(unchanged)"**, the execution of this scene will not change current status of this feature in the AC unit.*

⚠ ***Important:*** *When a scene is executed, Status_ Current Scene object shows the number of this scene. Any change in previous items does Status_ Current Scene show **"No Scene"**. Only changes on items marked as **"(unchanged)"** will not disable current scene.*

4.8 Switch-Off Timeouts Configuration dialog

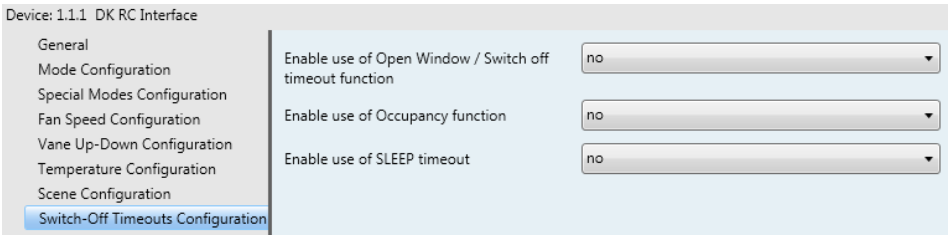
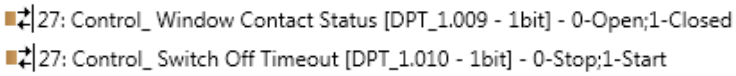


Figure 4.31 Default Switch-Off Timeouts Configuration dialog

All the parameters in this section are related with the timeout properties and communication objects.

4.8.1 Enable use of Open Window / Switch off timeout function

This parameter shows/hides the *Control_ Switch Off Timeout* communication object which lets you Start/Stop a timeout to switch off the indoor unit.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Control_ Switch Off Timeout* object and new parameters will appear. If a **"1"** value is sent to this object, and the indoor unit is already turned on, the switch-off timeout will begin. If a **"0"** value is sent to this object, the switch-off timeout will stop.

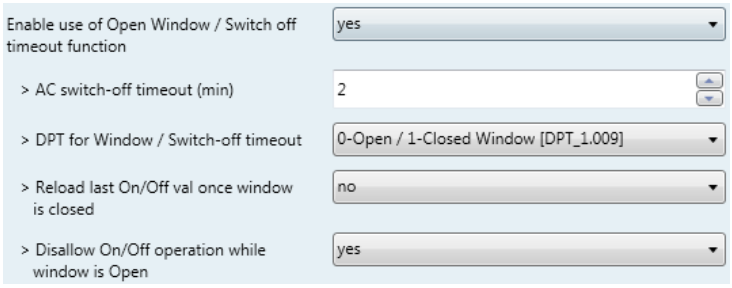


Figure 4.32 Parameter detail

➢ AC switch-off timeout (min)

This parameter lets you select how much time (in minutes) to wait before switching off the indoor unit.

➢ DPT for Window / Switch-off timeout

This parameter lets you choose between the datapoints **0-Open / 1-Closed Window [DPT_1.009]** and **0-Stop / 1-Start Timeout [DPT_1.010]** for the *Control_ Switch Off Timeout*.

➢ Disallow On/Off operation while window is Open

If set to **"no"**, On/Off commands while the window is open will be accepted.

- If a **"1"** value is sent to the *Control_ Switch Off Timeout* object the switch-off timeout period will begin again.
- If a **"0"** value is sent to the *Control_ Switch Off Timeout* object, no action will be performed.

If set to **"yes"**, On/Off commands, while the window is open, will be saved (but not applied). These commands will be used in the next parameter if set to **"yes"**.

➢ Reload last On/Off val once window is closed?

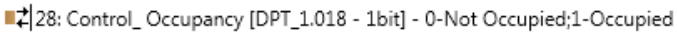
If set to **"no"**, once the switch-off timeout is stopped, any value will be reloaded.

If set to **"yes"**, once the switch-off timeout is stopped, the last On/Off value sent will be reloaded.

- If a **"1"** value is sent to the *Control_ Switch Off Timeout* object after the timeout period, the indoor unit will **turn on**.
- If a **"0"** value is sent to the *Control_ Switch Off Timeout* after the timeout period, no action will be performed.

4.8.2 Enable use of Occupancy function

This parameter shows/hides the *Control_ Occupancy* communication object which lets you apply different parameters to the indoor unit depending on the presence/no presence in the room.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Control_ Occupancy* object and new parameters will appear. If a **"1"** value is sent to this object (no room occupancy), the timeout will begin. If a **"0"** value is sent to this object, the timeout will stop.

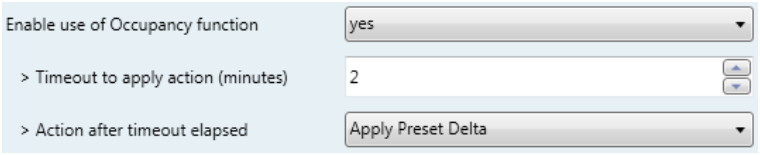


Figure 4.33 Parameter detail

➢ Timeout to apply action (minutes)

This parameter lets you choose how much time to wait (in minutes) before executing the action specified in the next parameter ("Action after timeout elapsed").

➢ Action after timeout elapsed

When **Switch-Off** is selected, once the timeout has elapsed, the indoor unit will be turned off.

When **Apply Preset Delta** is selected, once the timeout has elapsed, a delta temperature will be applied in order to save energy (decreasing the setpoint when in Heat mode, or increasing the setpoint when in Cool mode). Also new parameters will appear.

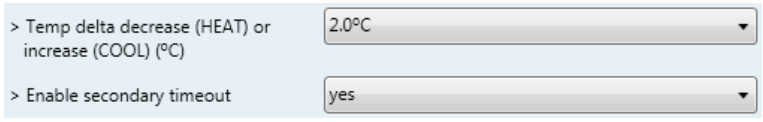


Figure 4.34 Parameter detail

- Temp delta decrease (HEAT) or increase (COOL) (ºC)

  This parameter lets configure the delta temperature (increase or decrease) that will be applied when the timeout has elapsed.

⚠ ***Important:*** *When there is occupancy again after the application of a delta, the same delta will be applied inversely. (i.e. In a room with AC in cool mode and 25ºC setpoint temperature, a* ***+2ºC*** *delta is applied after the occupancy timeout, setting the setpoint at 27ºC because there is no occupancy in the room. If the setpoint is raised to 29ºC during that period, when the room is occupied again, a* ***-2ºC*** *delta will be applied and the final setpoint temperature will then be 27ºC).*

- Enable secondary timeout

  If set to **"no"** nothing will be applied.
  If set to **"yes"**, a new timeout will be enabled and two new parameters will appear.

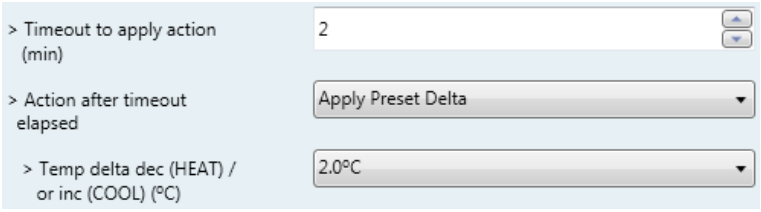


Figure 4.35 Parameter detail

  - Timeout to apply action (minutes)

    This parameter lets you choose how much time to wait (in minutes) before executing the action specified in the next parameter ("Action after timeout elapsed").

  - Action after timeout elapsed

    When **Switch-Off** is selected, once the timeouthas elapsed, the indoor unit will turn off.

    When **Apply Preset Delta** is selected, once the timeout configured is extinguished, a delta temperature will be applied (decreasing the setpoint when in Heat mode, or increasing the setpoint when in Cool mode). Also new parameters will appear.

    - Temp delta decrease (HEAT) or increase (COOL) (ºC)

      This parameter lets configure the delta temperature that will be applied when the timeout is extinguished.

    ⚠ ***Important:*** *When there is occupancy again after the application of a delta, the same delta will be applied inversely as explained above.*

- Disallow On/Off operation while not Occupied

  If set to **"no"**, On/Off commands while the window is open will be accepted.

- If a **"1"** value is sent to the *Control_ Occupancy* object the switch-off timeout period will begin again.
- If a **"0"** value is sent to the *Control_ Occupancy* object, no action will be performed.

If set to **"yes"**, On/Off commands while not occupied will be saved (but not applied). These commands will be used in the next parameter if set to **"yes"**.

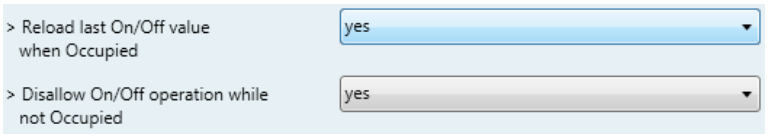


Figure 4.36 Parameter detail

➢ Reload last On/Off value when Occupied

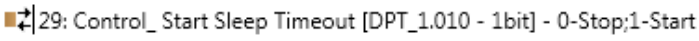
If set to **"no"**, once the switch-off timeout has elapsed, any value will be reloaded.

If set to **"yes"**, once the switch-off timeout has elapsed, the last On/Off value will be reloaded.

- If a **"1"** value is sent to the *Control_ Occupancy* object after the timeout period, the indoor unit will **turn on**.
- If a **"0"** value is sent to the *Control_ Occupancy* after the timeout period no action will be performed.

4.8.3 Enable use of SLEEP timeout

This parameter shows/hides the *Control_ Start Sleep Timeout* communication object which lets you start a timeout to automatically turn off the indoor unit.



- If set to **"no"** the object will not be shown.
- If set to **"yes"** the *Control_ Start Sleep Timeout* object and a new parameter will appear. If a **"1"** value is sent to this object the switch-off timeout will begin. If a **"0"** value is sent to this object, the switch-off timeout will stop.

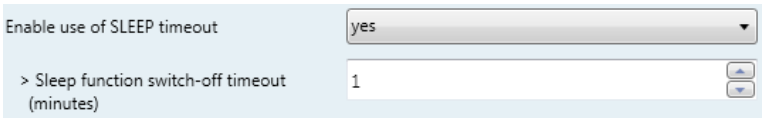


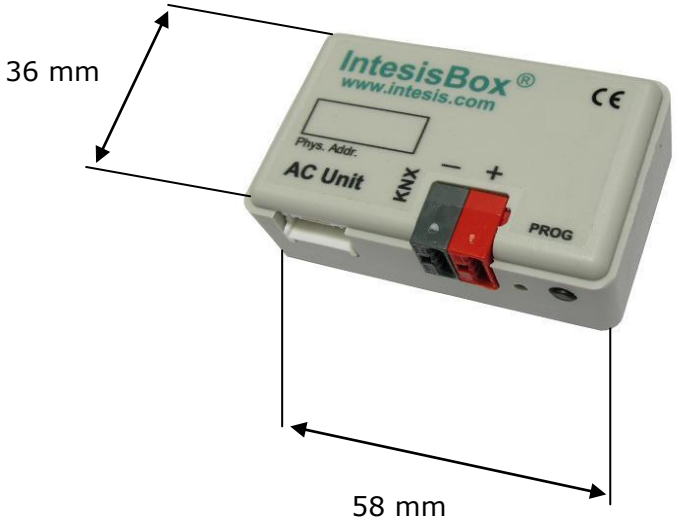
Figure 4.37 Parameter detail

➢ Timeout to apply action (minutes)

This parameter lets you select how much time (in minutes) to wait before switching off the AC unit.

5 Specifications

Dimensions:	59 X 36 X 21 mm
Weight:	42 g
KNX current consumption:	5 mA
Operating Temperature:	-25 . . . 85ºC
Stock Temperature:	-40 . . . 85ºC
Isolation voltage:	4000 V



6 AC Unit Types compatibility.

A list of Midea indoor unit model references compatible with MD-AC-KNX-1 and their available features can be found in:

http://www.intesis.com/pdf/IntesisBox_MD-AC-xxx-1_AC_Compatibility.pdf

7 Error Codes

Error Code KNX Object	Error in Remote Controller	Error Name
1	E0	Phase error or error in the pahse sequence
2	E1	Communication error
3	E2	T1 sensor error
4	E3	T2A sensor error
5	E4	T2B sensor error
6	E5	T3 temperaature and T4 temperature Compressor discharge temperature sensors error
7	E6	Zero cross error detection
8	E7	EEPROM memory error
9	E8	Indoor fan speed out of control
10	E9	Communication error between the main panel and the visualization panel
11	EA	Compressor's current overload error (4 times)
12	EB	Inverter module protection
13	EC	Cooling error
14	ED	Outdoor unit foult protection
15	EE	Water level fault detection
16	EF	Other errors
101	P0	Vaporizer temperature protection
102	P1	Thawing or cold air protection
103	P2	Condensor high temperatures protection
104	P3	Compressor temperature protection
105	P4	Evacuation duct temperature protection
106	P5	Discharge high pressure protection
107	P6	Discharge low pressure protection
108	P7	Current overload or underload protection
109	P8	Compressor's current overload protection
110	P9	Reserved
111	PA	Reserved
112	PB	Reserved
113	PC	Reserved
114	PD	Reserved
115	PE	Reserved
116	PF	Other protection mesures
-1	-	Communication error between MD-AC-KNX-1 and Indoor Unit

In case you detect an error code not listed, contact your nearest Midea technical support service fro more information on the error meaning.

Appendix A – Communication Objects Table

SECTION	OBJECT NUMBER	NAME	LENGTH	DATAPOINT TYPE		FLAGS				FUNCTION
				DPT_NAME	DPT_ID	R	W	T	U	
On/Off	0	Control_ On/Off	1 bit	DPT_Switch	1.001		W	T		0 - Off; 1-On
Mode	1	Control_ Operating Mode	1 byte	DPT_HVACMode	20.102		W	T		0 - Auto; 1 - Com; 2 - Stan; 3 - Eco; 4 – Pro
	2	Control_ Mode	1 byte	DPT_HVACControl	20.105		W	T		0 - Auto; 1 - Heat; 3 - Cool; 9 - Fan; 14 - Dry
	3	Control_ Mode Cool/Heat	1 bit	DPT_Cool/Heat	1.100		W	T		0 - Cool; 1 – Heat
	4	Control_ Mode Cool & On	1 byte	DPT_Scaling	5.001		W	T		0% - Off; 0.1%-100% - On + Cool
	5	Control_ Mode Heat & On	1 byte	DPT_Scaling	5.001		W	T		0% - Off; 0.1%-100% - On + Heat
	6	Control_ Mode Auto	1 bit	DPT_Bool	1.002		W	T		1 - Auto
	7	Control_ Mode Heat	1 bit	DPT_Bool	1.002		W	T		1 - Heat
	8	Control_ Mode Cool	1 bit	DPT_Bool	1.002		W	T		1 - Cool
	9	Control_ Mode Fan	1 bit	DPT_Bool	1.002		W	T		1 - Fan
	10	Control_ Mode Dry	1 bit	DPT_Bool	1.002		W	T		1 - Dry
	11	Control_ Mode +/-	1 bit	DPT_Step	1.007		W			0 - Decrease; 1 - Increase
		Control_ Mode +/-	1 bit	DPT_UpDown	1.008		W			0 - Up; 1 - Down
Fan Speed	12	Control_ Fan Speed / 2 Speeds	1 byte	DPT_Scaling	5.001		W	T		0%-74% - Speed 1; 75%-100% - Speed 2
		Control_ Fan Speed / 3 Speeds	1 byte	DPT_Scaling	5.001		W	T		0%-49% - Speed 1; 50%-83% - Speed 2; 84%-100% Speed 3
		Control_ Fan Speed / 2 Speeds	1 byte	DPT_Enumerated	5.010		W	T		1 - Speed 1; 2 - Speed 2
		Control_ Fan Speed / 3 Speeds	1 byte	DPT_Enumerated	5.010		W	T		1 - Speed 1; 2 - Speed 2; 3 Speed 3

	13	Control_ Fan Speed Man/Auto	1 bit	DPT_Bool	1.002		W	T		0 – Manual; 1 – Auto
	14	Control_ Fan Speed 1	1 bit	DPT_Bool	1.002		W	T		1 - Fan Speed 1
	15	Control_ Fan Speed 2	1 bit	DPT_Bool	1.002		W	T		1 - Fan Speed 2
	16	Control_ Fan Speed 3	1 bit	DPT_Bool	1.002		W	T		1 - Fan Speed 3
	17	Control_ Fan Speed +/-	1 bit	DPT_Step	1.007		W	T		0 - Decrease; 1 - Increase
		Control_ Fan Speed +/-	1 bit	DPT_UpDown	1.008		W	T		0 - Up; 1 - Down
Vanes	**18**	Control_ Vanes U-D Swing	1 bit	DPT_Bool	1.002		W	T		0 - Off; 1 - Swing
Temperature	**19**	Control_ Setpoint Temperature	2 byte	DPT_Value_Temp	9.001		W	T		17ºC to 30ºC
	20	Control_ Setpoint Temp +/-	1 bit	DPT_Step	1.007		W			0 - Decrease; 1 - Increase
		Control_ Setpoint Temp +/-	1 bit	DPT_UpDown	1.008		W			0 - Up; 1 - Down
	21	Control_ Ambient Temperature	2 byte	DPT_Value_Temp	9.001		W	T		ºC value in EIS5 format
Timeout	**22**	Control_ Window Contact Status	1 bit	DPT_OpenClose	1.009		W	T		0 - Open; 1 - Closed
		Control_ Switch Off Timeout	1 bit	DPT_Start	1.010		W	T		0 - Stop; 1 - Start
	23	Control_ Occupancy	1 bit	DPT_Occupancy	1.018		W	T		0 - Not Occupied; 1 - Occupied
	24	Control_ Sleep Timeout	1 bit	DPT_Start	1.010		W	T		0 - Stop; 1 - Start
Locking	**25**	Control_ Lock Remote Control	1 bit	DPT_Bool	1.002		W	T		0 - Unlocked; 1 - Locked
	26	Control_ Lock Control Objects	1 bit	DPT_Bool	1.002		W	T		0 - Unlocked; 1 - Locked
Special Modes	**27**	Control_ Power Mode	1 bit	DPT_Start	1.010		W	T		0 - Stop; 1 - Start
	28	Contorl_ Econo Mode	1 bit	DPT_Start	1.010		W	T		0 - Stop; 1 - Start
	29	Control_ Additional Heat	1 bit	DPT_Start	1.010		W	T		0 - Stop; 1 - Start
	30	Control_ Additional Cool	1 bit	DPT_Start	1.010		W	T		0 - Stop; 1 - Start

Scenes	**31**	Control_ Save/Exec Scene	1 byte	DPT_SceneControl	18.001		W			0 to 4 - Exec. Scene 1 to 5; 128 to 132 - Save Scene 1 to 5
	32	Control_ Store Scene1	1 bit	DPT_Bool	1.002		W			1 - Store Scene
	33	Control_ Store Scene2	1 bit	DPT_Bool	1.002		W			1 - Store Scene
	34	Control_ Store Scene3	1 bit	DPT_Bool	1.002		W			1 - Store Scene
	35	Control_ Store Scene4	1 bit	DPT_Bool	1.002		W			1 - Store Scene
	36	Control_ Store Scene5	1 bit	DPT_Bool	1.002		W			1 - Store Scene
	37	Control_ Execute Scene1	1 bit	DPT_Bool	1.002		W			1 - Execute Scene
	38	Control_ Execute Scene2	1 bit	DPT_Bool	1.002		W			1 - Execute Scene
	39	Control_ Execute Scene3	1 bit	DPT_Bool	1.002		W			1 - Execute Scene
	40	Control_ Execute Scene4	1 bit	DPT_Bool	1.002		W			1 - Execute Scene
	41	Control_ Execute Scene5	1 bit	DPT_Bool	1.002		W			1 - Execute Scene
On/Off	**42**	Status_ On/Off	1 bit	DPT_Switch	1.001	R		T		0 - Off; 1-On
Mode	**43**	Status_ Operating Mode	1 byte	DPT_HVACMode	20.102	R		T		0 - Auto; 1 - Com; 2 - Stan; 3 - Eco; 4 – Pro
	44	Status_ Mode	1 byte	DPT_HVACContrMode	20.105	R		T		0 - Auto; 1 - Heat; 3 - Cool; 9 - Fan; 14 - Dry
	45	Status_ Mode Cool/Heat	1 bit	DPT_Heat/Cool	1.100	R		T		0 - Cool; 1 - Heat
	46	Status_ Mode Auto	1 bit	DPT_Bool	1.002	R		T		1 - Auto
	47	Status_ Mode Heat	1 bit	DPT_Bool	1.002	R		T		1 - Heat
	48	Status_ Mode Cool	1 bit	DPT_Bool	1.002	R		T		1 - Cool
	49	Status_ Mode Fan	1 bit	DPT_Bool	1.002	R		T		1 - Fan
	50	Status_ Mode Dry	1 bit	DPT_Bool	1.002	R		T		1 - Dry
	51	Status_ Mode Text	14 byte	DPT_String_8859_1	16.001	R		T		ASCII String

Fan Speed	**52**	Status_ Fan Speed / 2 Speeds	1 byte	DPT_Scaling	5.001	R		T		50% - Speed 1; 100% - Speed 2
		Status_ Fan Speed / 3 Speeds	1 byte	DPT_Scaling	5.001	R		T		33% - Speed 1; 67% - Speed 2; 100% - Speed 3
		Status_ Fan Speed / 2 Speeds	1 byte	DPT_Enumerated	5.010	R		T		1 - Speed 1; 2 - Speed 2
		Status_ Fan Speed / 3 Speeds	1 byte	DPT_Enumerated	5.010	R		T		1 - Speed 1; 2 - Speed 2; 3 - Speed 3
	53	Status_ Fan Speed Man/Auto	1 bit	DPT_Bool	1.002	R		T		0 – Manual; 1 – Auto
	54	Status_ Fan Speed 1	1 bit	DPT_Bool	1.002	R		T		1 - Speed 1
	55	Status_ Fan Speed 2	1 bit	DPT_Bool	1.002	R		T		1 - Speed 2
	56	Status_ Fan Speed 3	1 bit	DPT_Bool	1.002	R		T		1 - Speed 3
	57	Status_ Fan Speed Text	14 byte	DPT_String_8859_1	16.001	R		T		ASCII String
Vanes	**58**	Status_ Vane Up-Down Swing	1 bit	DPT_Bool	1.002	R		T		0 - Stop; 1 - Swing
Temperature	**59**	Status_ AC Setpoint Temp	2 byte	DPT_Value_Temp	9.001	R		T		16ºC to 32ºC
	60	Status_ AC Ambient Ref Temp	2 byte	DPT_Value_Temp	9.001	R		T		ºC value in EIS5 format
Error	**61**	Status_ Error/Alarm	1 bit	DTP_Alarm	1.005	R		T		0 - No Alarm; 1 - Alarm
	62	Status_ Error Code	2 byte	Enumerated		R		T		0 - No Error; Any other see user's manual
	63	Status_ Error Text code	14 byte	DPT_String_8859_1	16.001	R		T		2 char MD Error; Empty - none
Special Modes	**64**	Status_ Power Mode	1 bit	DPT_Switch	1.001	R		T		0 - Off; 1-On
	65	Status_ Econo Mode	1 bit	DPT_Switch	1.001	R		T		0 - Off; 1-On
	66	Status_ Additional Heat	1 bit	DPT_Switch	1.001	R		T		0 - Off; 1-On
	67	Status_ Additional Cool	1 bit	DPT_Switch	1.001	R		T		0 - Off; 1-On
Counter	**68**	Status_ Operation Hour Counter	2 byte	DPT_Value_2_Ucount	7.001	R		T		Number of operating hours
Scene	**69**	Status_ Current Scene	1 byte	DPT_SceneNumber	17.001	R		T		0 to 4 - Scene 1 to 5; 63 - No Scene