KNX Module
Universal Interface Adapter
for HVAC Systems
Attention

1. During the HVAC system automatic operations the CoolMasterNet must be disconnected from the system.
   Examples of the automatic operations are:
   · Test
   · Reset
   · Automatic charge
2. Do not change the original positions of the quadruple dip switch S inside CoolMasterNet when using it with VRV/VRF systems.

WEEE Directive & Product Disposal

At the end of its serviceable life, this product should not be treated as household or general waste. It should be handed over to the applicable collection point for the recycling of electrical and electronic equipment, or returned to the supplier for disposal.
Table of Contents

1 Overview 4

2 Configuration 5
   2.1 KNX Module activation .......................................................... 5
   2.2 KNX Module settings ............................................................... 5

3 KNX Project 7
   3.1 Group Addresses .................................................................. 7
   3.2 KNX group command ............................................................ 7
   3.3 KNX Module functions .......................................................... 8
   3.4 Examples of grouping ........................................................... 9

4 KNX Module commands reference 11
1 Overview

Physical connection of CoolAutomation devices: CoolMasterNet or CooLinkNet to the KNX bus with TP physical layer is made with "CoolAutomation KNX expansion board", further referenced as "KNX Module".

KNX Module is pre-installed inside CoolMasterNet or CooLinkNet body by CoolAutomation if KNX expansion option was ordered together with a CoolAutomation device.
## 2 Configuration

### 2.1 KNX Module activation

KNX Module has to be activated by assigning appropriate communication Line of the CoolAutomation device for the KNX functionality. In CoolMasterNet it has to be Line L3 and in CooLinkNet it has to be Line L5. To check if KNX Module is already activated, `line` command should be used:

<table>
<thead>
<tr>
<th>CoolMasterNet</th>
<th>CooLinkNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>line</code></td>
<td><code>line</code></td>
</tr>
</tbody>
</table>

```plaintext
CoolMasterNet
>
L1: DK Master U00/G00 myid:0B
Tx:2/2 Rx:2/2 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
L2: Unused
Tx:0/0 Rx:0/0 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
L3: KNX Physical Address:Not Assigned
Tx:0/0 Rx:0/0 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
L4: Unused
Tx:0/0 Rx:0/0 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
L5: Unused
Tx:0/0 Rx:0/0 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
L6: Unused
Tx:0/0 Rx:0/0 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
L7: Unused
Tx:0/0 Rx:0/0 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
L8: Unused
Tx:0/0 Rx:0/0 TO:0/0 CS:0/0 Col:0/0 NAK:0/0
OK
```

If for some reason KNX Module is not activated, it can be done with below command:

<table>
<thead>
<tr>
<th>CoolMasterNet</th>
<th>CooLinkNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>line type L3 KNX</code></td>
<td><code>line type L5 KNX</code></td>
</tr>
</tbody>
</table>

```plaintext
CoolMasterNet
>
OK, Boot Required!
```

<table>
<thead>
<tr>
<th>CooLinkNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>line type L5 KNX</code></td>
</tr>
</tbody>
</table>

```plaintext
OK, Boot Required!
```

### 2.2 KNX Module settings

- Setting a number of supported group addresses.

By default KNX Module supports up to 128 group addresses. If that amount is not sufficient it can be increased (or decreased) with below command:

```plaintext
>knx ram 200
OK, Boot Required!
```

This will increase a maximal number of group addresses supported by KNX Module to 200. The absolute maximal number of supported group addresses is 1024.

- Setting a physical address of the KNX Module.

Physical address (also referenced as Individual Address in KNX specifications) has a following format:
The physical address has no significance during normal operation of the installation. For KNX Module, physical address can be left unassigned or, if required, it can be set with command like:

```
> knx addr 10/2/45
OK
>
```

An individual address must be unique within a KNX installation.
3 KNX Project

3.1 Group Addresses

KNX bus is implemented as a decentralized system. Decentralized management is implemented within the devices, they are communicating directly to each other without recourse to hierarchy or network supervisory. The communication between KNX communication objects (sensors and actuators) is made with telegrams sent to group addresses. The group addressing is based on the exchange of data coded with common rules between communication objects. Actuators can listen to several group addresses. Sensors however can only send one group address per telegram.

ETS tool should be used to assign group addresses for communication objects provided by KNX devices used in project. CoolMasterNet and CooLinkNet KNX Modules do not require ETS interaction. CoolAutomation devices are configured by internal set of commands described later in this document.

When setting the group address via ETS, a “2-level” (main group/ subgroup) or “3-level” structure (main group/middle group/subgroup) can be selected. The level structure can be changed in the project properties of each individual project. The group address 0/0/0 is reserved for so-called broadcast messages (telegrams to all available bus devices).

- Group Address 2 level Main/Sub

<table>
<thead>
<tr>
<th>Main Group: 5bit</th>
<th>Sub Group: 11bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M M M M M S S S S S S S</td>
<td></td>
</tr>
</tbody>
</table>

- Group Address 3 level Main/Middle/Sub

<table>
<thead>
<tr>
<th>Main Group: 5bit</th>
<th>Middle Group: 3bit</th>
<th>Sub Group: 8bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M M M M M Mi Mi Mi S S S S S S</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 KNX group command

To link KNX group objects, defined with ETS tool, with CoolMasterNet or CooLinkNet functions, ‘knx group’ command should be used.

```
knx group [ <GA> <func> <direction> <UID> ]
```

- `<GA>`: KNX group address. M/S or M/Mi/S structure formats are supported
- `<func>`: Required CoolMasterNet or CooLinkNet function (see KNX functions).
- `<direction>`: Data direction, encoded as:
  - `<`: Into KNX Module from Group Object with W - Write direction flag
  - `>`: From KNX Module to Group Object with R - Read direction flag
- `<UID>`: Indoor Unit, in same format as reported by `ls` command

KNX group command binds between KNX Group Object with given group address - `<GA>` and specific function of CoolMasterNet or CooLinkNet that will be applied to the Indoor Unit(s) with given `<UID>`. Groups are stored in CoolMasterNet or CooLinkNet non volatile memory. The maximal number of groups is configurable (see KNX Module settings).

Other formats of the ‘knx group’ command invocations provide options to list already defined groups or delete specific or all groups.

List KNX groups:
```
> knx group
G000:10/0/1 [ On/Off ] < L1.002
```

www.coolautomation.com © 2019 CoolAutomation LTD.
#### 3.3 KNX Module functions

The `<func>` parameter of the `knx group` command defines both: function associated with a group and a datapoint type of the information that will be transferred in corresponding KNX telegram. Functions supported by KNX Module are listed below.

<table>
<thead>
<tr>
<th><code>&lt;func&gt;</code></th>
<th>Function description</th>
<th>Datapoint Type</th>
<th>Dir.</th>
<th>Min. FW Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>onoff</td>
<td>ON/OFF control or status. 0-OFF, 1-ON</td>
<td>{1.001} DPT_Switch B1</td>
<td>&lt; &gt;</td>
<td>Any</td>
</tr>
<tr>
<td>ST</td>
<td>Set point temperature °C</td>
<td>{9.001} DPT_Value_Temp F16</td>
<td>&lt; &gt;</td>
<td></td>
</tr>
<tr>
<td>RT</td>
<td>Room temperature °C</td>
<td>{9.001} DPT_Value_Temp F16</td>
<td>&lt; &gt;</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Mode: Cool/Heat. 0-Cool, 1-Heat</td>
<td>{1.001} DPT_Switch B1</td>
<td>&lt; &gt;</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>Mode: 0-Cool, 1-Heat, 2-Auto, 3-Dry, 5-Fan, 8 - HRV Auto 9 - HRV Bypass 10 - HRV Heat Exchange 11 - HRV Normal Mode 12 - HRV Sleep Mode</td>
<td>{5.010} DPT_Value_1_Ucount U8</td>
<td>&lt; &gt;</td>
<td>0.2.8</td>
</tr>
<tr>
<td>Fan</td>
<td>Fan speed: 0-Low, 1-Medium, 2-High, 3-Auto, 4-Top, 5-Very Low, 6-Super High, 7-HRV Super High, 8-HRV Low Freshup, 9-HRV High Freshup</td>
<td>{5.010} DPT_Value_1_Ucount U8</td>
<td>&lt; &gt;</td>
<td>0.6.6</td>
</tr>
<tr>
<td>Fstep</td>
<td>Skip to next fan speed</td>
<td>{1.001} DPT_Switch B1</td>
<td>&lt;</td>
<td>0.2.1</td>
</tr>
<tr>
<td>F8</td>
<td>Fan speed count</td>
<td>{5.010} DPT_Value_1_Ucount U8</td>
<td>&lt; &gt;</td>
<td>0.2.1</td>
</tr>
<tr>
<td>F%</td>
<td>Fan speed scaled</td>
<td>{5.001} DPT_Scale</td>
<td>&lt; &gt;</td>
<td>0.2.3</td>
</tr>
<tr>
<td>E16</td>
<td>HVAC Error (failure) code (0 - no failure)</td>
<td>{7,1} DPT_Value_2_Ucount U16</td>
<td>&gt;</td>
<td>0.4.9</td>
</tr>
<tr>
<td>E</td>
<td>HVAC Error (failure) indication: 0-no failure, 1-failure exists</td>
<td>{1,1} DPT_Switch B1</td>
<td>&gt;</td>
<td>0.4.9</td>
</tr>
<tr>
<td>inhb</td>
<td>Inhibit (see notes below)</td>
<td>{1,1} DPT_Switch B1</td>
<td>&lt;</td>
<td>0.4.9</td>
</tr>
<tr>
<td>filter</td>
<td>Filter cleaning indication</td>
<td>{1,1} DPT_Switch B1</td>
<td>&lt; &gt;</td>
<td>0.4.9</td>
</tr>
<tr>
<td>Swing</td>
<td>Louver position/swing control:</td>
<td>{5.010} DPT_Value_1_Ucount U8</td>
<td>&lt; &gt;</td>
<td>0.4.9</td>
</tr>
</tbody>
</table>
Notes:

- **Fstep** function takes current fan speed and increases it in one step (regardless to the value sent in KNX telegram). If current fan speed is maximal supported by related Indoor Unit, lowest possible fan speed will be taken. For example if Indoor Unit supports Low, Medium and High fan speeds **Fstep** will change fan speed from Low to Medium, from Medium to High or from High to Low. This function has only one direction - from KNX device to the KNX Module.

- **RT In** direction (to the KNX Module), upon reception of the corresponding KNX telegram datapoint value will be used as "Feed Temperature" - Ambient Temperature suggestion for related Indoor Unit. If value is zero CoolMasterNet will stop suggesting Ambient Temperature. Suggested Ambient Temperature is not persistent and is lost (set to zero) after power reset of CoolMasterNet. It is not guaranteed that the Ambient Temperature reported by Indoor Unit will be equal to the suggested Ambient Temperature provided with **RT In** function. The **RT In** function is supported for the following AC types:

<table>
<thead>
<tr>
<th>AC Type</th>
<th>FW Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td></td>
</tr>
<tr>
<td>PBM</td>
<td></td>
</tr>
<tr>
<td>SI</td>
<td>0.4.7</td>
</tr>
</tbody>
</table>

- **F8** function takes all supported fan speeds of the related Indoor Unit and gives them a numbers starting from 1 (one). In < direction (to the KNX Module), upon reception of the corresponding KNX telegram datapoint value will be translated into fan speed. In > direction (from the KNX Module) current fan speed number will be sent in KNX telegram.

- **F%** function is similar to **F8** function except of the fan speed encoding. In case of **F%**, encoding is scaled in % according to the below table.

<table>
<thead>
<tr>
<th>Supported Fan Speeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
</tr>
<tr>
<td>0%</td>
</tr>
<tr>
<td>0%</td>
</tr>
<tr>
<td>0%</td>
</tr>
</tbody>
</table>

- **inhb** function activates or deactivates (according to datapoint value 1 or 0) inhibit of the indoor unit ON operation. Upon inhibit activation current ON/OFF status of the indoor unit is stored and indoor unit is unconditionally turned OFF after about 8s delay. Indoor unit will be forced OFF until inhibit deactivation. On inhibit deactivation indoor unit will be turned ON if ON status was previously stored at inhibit activation. This function is oriented to work with window sensor to prevent HVAC operation while window is open.

- **Swing** function may have no effect if requested louver position is not supported by indoor unit. Not all indoor units support louver position options listed above or have louver position control at all. Louver control is a capability of the specific indoor unit type.

### 3.4 Examples of grouping

knx group 10/0/1 onoff < L1.101

Indoor Unit L1.101 will be switched ON or OFF, once KNX telegram with destination group address 10/0/1 and 1-bit datapoint type {1.001} is sent by KNX device like thermostat or push-button. Value of "0" will switch the Indoor Unit OFF and value of "1" - ON.

knx group 10/0/2 onoff > L1.101

OK
Upon ON/OFF status change of the Indoor Unit L1.101, KNX telegram with destination group address 10/0/2 will be sent by KNX Module. Telegram will contain 1-bit datapoint type with value "0" - if Indoor Unit status is OFF and "1" - if ON. Note how the direction characters '>' and '<' work.

```plaintext
>knx group 10/0/3 ST < L1.101
OK
```
Set point temperature of the Indoor Unit L1.101 will be changed by KNX Module upon reception of the KNX telegram to destination group address 10/0/0/3. Temperature value is encoded as {9.001} datapoint type.

```plaintext
>knx group 10/0/4 ST > L1.101
OK
```
In this case change of the set point temperature of the Indoor Unit L1.101 will be reported by KNX Module that will send KNX telegram to group address 10/0/4 with {9.001} datapoint type.
4 KNX Module commands reference

- **knx** - Query KNX Module status

  ```
  >knx
  KNX Line : Connected
  KNX RAM : 10/128
  OK
  >
  ```

  KNX Line status can be "Connected" or "Disconnected". RAM usage reflects a number of defined group addresses and a total supported number of group addresses.

- **knx addr <A/ L/ D>** - Set KNX Module physical address.

  ```
  >knx addr 10/2/45
  OK
  ```

- **knx ram <N>** - Set number of supported group addresses

  ```
  >knx ram 200
  OK, Boot Required!
  ```

- **knx group** - List defined KNX groups

  ```
  >knx group
  G000:10/0/1 [ On/Off ] < L1.002
  G001:10/0/2 [ On/Off ] > L1.002
  G002:10/0/3 [ Set temperature ] > L1.002
  G003:10/0/4 [ Set temperature ] < L1.002
  G004:10/0/5 [ Mode ] > L1.002
  G005:10/0/6 [ Mode ] < L1.002
  OK
  ```

- **knx group <GA> <func> <direction> <U|D>** - Define KNX group

  ```
  >knx group 10/0/3 ST < L1.101
  OK
  ```

- **knx group <N>** - Delete KNX group

  ```
  >knx group -2
  OK
  ```

- **knx group del all** - Delete all KNX groups
>knx group delall
OK

- **knx funcs** - List supported functions

```plaintext
>knx funcs

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Data Type</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>onoff</td>
<td>On/Off</td>
<td>DT:{1.001}</td>
<td>[&lt;&gt;]</td>
</tr>
<tr>
<td>ST</td>
<td>Set temperature</td>
<td>DT:{9.001}</td>
<td>[&lt;&gt;]</td>
</tr>
<tr>
<td>RT</td>
<td>Room temperature</td>
<td>DT:{9.001}</td>
<td>[&lt;&gt;]</td>
</tr>
<tr>
<td>M</td>
<td>Mode</td>
<td>DT:{1.001}</td>
<td>[&lt;&gt;]</td>
</tr>
<tr>
<td>Fstep</td>
<td>Fan speed step</td>
<td>DT:{1.001}</td>
<td>[&lt;&gt;]</td>
</tr>
<tr>
<td>F8</td>
<td>Fan speed count</td>
<td>DT:{5.010}</td>
<td>[&lt;&gt;]</td>
</tr>
<tr>
<td>F%</td>
<td>Fan speed scale</td>
<td>DT:{5.001}</td>
<td>[&lt;&gt;]</td>
</tr>
</tbody>
</table>

OK
```