KNX protocol

KNX protocol

Supported device types and versions
Communication line configuration
Line protocol parameters
Communication station configuration
I/O tag configuration
Literature
Changes and modifications
Document revisions

Supported device types and versions

The KNX protocol is an open standard used in building automation. The KNX implementation uses the Falcon SDK developed by the KNX association. Since this SDK is developed in C#, the connection to the D2000 KOM process is handled by a separate module d2knx, which by default is located in the *pr otdll* directory and communicates with the D2000 KOM process via a TCP connection.
For each communication line on which there are stations with KNX protocol, it is necessary to run one instance of the d2knx module, which requires a separate KNX interface. The KNX interface can be connected via the USB interface or can be accessed via a network based on its IP address.

Note on the d2knx module:

The d2knx module is an application that needs ".NET Core 3.1" installed to run. We recommend that you test its functionality by running it manually from the command line.
When started without parameters, it displays help and a list of available KNX USB devices and then exits.
The required parameters are:

- IP address - the address on which d2knx listens (e.g. 127.0.0.1 or 0.0.0.0 or the specific IP address of the computer where the d2knx module is running, e.g. 172.16.0.1)
- port - TCP port on which d2knx is listening
- debug - an optional parameter that activates output of auxiliary debug info

Examples of manually running d2knx:

- d2knx 127.0.0.1 4011 debug
- d2knx 0.0.0.0 4012

Example output (if start without parameters):

```
EIB/KNX interface module
(c) 2020 Ipesoft
Usage : knx ListenIP ListenPort [debug]
Example: knx 127.0.0.1 4011
Available interfaces:
Device 0 Path \\?\hid#vid_16d0&pid_0490#6&34ad9346&0&0000#{4d1e55b2-f16f-11cf-88cb-001111000030}, Name Tapko USB Interface
```

Communication line configuration

- Communication line category: TCP/IP-TCP.
- The IP address is the address where the d2knx module listens. If the d2knx module is started by the D2000 KOM process, the recommended address is 127.0.0.1 (localhost). It is possible to run the d2knx module on a remote computer, in which case the IP address of the remote computer must be configured.
  Note: it is possible to configure multiple IP addresses (separated by a comma or semicolon). Thus, in redundant D2000 systems, it is possible to configure the connection of the D2000 KOM process, e.g. to independently running d2knx modules (each of which is running on one of the application servers), which ensures communication functionality in case of failure of the KNX USB adapter on one server without the need to switch D2000 redundancy.
- The port number is the TCP port number on which the d2knx module listens. You can select any free port (1-65535).
- The line number is unused, set the value to 0.

Line protocol parameters

A dialog window of communication line configuration - **Protocol parameters** tab.
They influence some optional protocol parameters.

The following line protocol parameters are defined:

Parameter	Meaning	Unit / Size	Default value

Parameter	Meaning	Unit / Size	Default value
KNX Driver Program	Configuration of starting the d2knx module by the D2000 KOM process. If this parameter is empty, the d2knx module will not be started (another starting mechanism is assumed). The parameter contains the following symbolic names: • # PROTDIR # - it will be replaced by the path to the *protdll* directory in the D2000 installation, where d2knx is located by default (e.g. D:\D2000\D2000.EXE*protdll*) • # BINDIR # - it will be replaced with the path to the *bin* directory (*bin64* in the 64-bit version of D2000 on Windows) in the D2000 installation • # HOST # - it will be replaced by the IP address configured on the line (in case of the configuration of several IP addresses, these will be used alternately) • # PORT # - it will be replaced by the TCP port number configured on the line Note: The *protdll* directory also contains sample startup scripts for Windows (*d2knx.bat*) and Linux/Raspberry PI (*d2knx*).	-	#PROTDIR#d2knx #HOST# #PORT#
KNX Interface Type	Type of KNX adapter that the d2knx module opens and with which it communicates via the KNX bus: • *USB interface by position* - from the list of available KNX USB devices, the device at the position defined by the KNX Interface ID parameter (number 0, 1, 2, etc.) is selected • *USB interface by name* - a device whose name contains the text entered by the KNX Interface ID parameter (e.g. "Tapco") is selected from the list of available KNX USB devices • *USB interface by path* - from the list of available KNX USB devices, a device whose path contains the text specified by the KNX Interface ID parameter is selected (e.g. "4d1e55b2") • *IP interface (IP address, port, protocol, NAT)* - the device whose IP address is in the KNX Interface ID parameter is used	-	USB interface by position
KNX Interface ID	Identification of the KNX adapter to be opened, which depends on the selected KNX interface type - see the description of the KNX Interface Type parameter. This can be the serial number of the USB interface, part of its name or path, or the IP address of the KNX router.	-	0
IP Interface Port	If KNX Interface ID = "I*P interface*", then this parameter specifies the port number (TCP or UDP) on which the KNX router communicates. The default port number is 3671.	-	3671
IP Interface Protocol	If KNX Interface ID = "I*P interface*", then this parameter indicates whether UDP or TCP protocol is used for communication with the KNX router: • Automatic - automatic detection (default) • UDP - UDP protocol will be used • TCP - TCP protocol will be used	Automatic UDP TCP	Automatic
IP Interface NAT	If KNX Interface ID = "I*P interface*" and communication goes via the UDP protocol, this parameter specifies whether address translation (NAT) is to be used for communication.	YES /NO	NO
KNX Interface Address (x.y.z)	KNX address that can be set on the KNX interface after opening it. If not specified, the existing KNX interface address is used. The KNX address has the format area.line.device (e.g. 1.3.99) where the area is 0..15, the line is 0..15, the device is 0..255.	-	-
Security Keys	The name of the security key file. If secure group communication is used, this file (so-called keyring file - * .knxkeys) can be exported from the ETS configuration tool, which is available on the KNX asociácie's website. The parameter contains the following symbolic names: • # APPDIR # - it will be replaced by the path to the application directory (e.g. D:\D2000\D2000.APP\myapp\) Examples: #APPDIR#myfile.knxkeys D:\keys.knxkeys	-	-
Security Password	A password to access the security key file. Note: If secure communication is used, both *Security Keys* and *Security Password* must be specified.	-	-
Driver Debug	This parameter activates the d2knx module debugging information. Zapnutie ladiacich informácii modulu d2knx.	YES /NO	NO
Full Debug	This parameter activates the debugging information about the read/written values.	YES /NO	NO

Communication station configuration

- Communication protocol: **KNX Protocol**.
- Station address:
    - *GROUP* - The I/O tags on a station with the *GROUP* address will receive all values of the *Group address* type values from the communication.
    - *area.line.device* - Address of a specific KNX device (e.g. 1.3.99), where the area is 0..15, the line is 0..15, the device is 0..255. I/O tags on a station with a specific address will only receive values if they have been received from a KNX device with a specified address.

Note: it is possible to configure several stations with the same address (with a specific address or with a *GROUP* address), e.g. to set various time parameters for reading individual I/O tags.

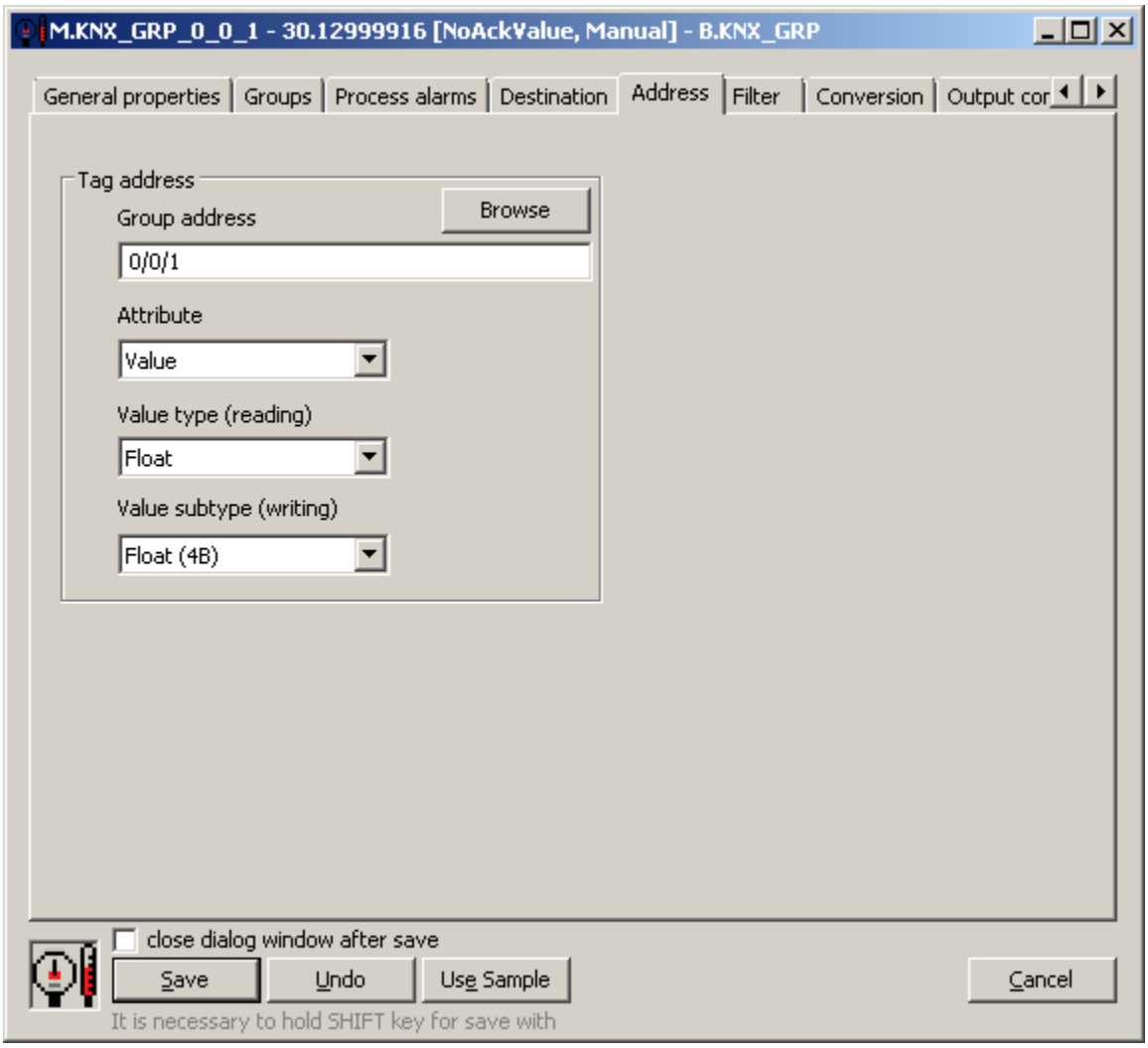
Communication station protocol parameters

The following station parameters can be set:

Parameter	Meaning	Unit / Size	Default value
Read Group Addresses	Method of explicit reading of I/O tags of type *Group address*: • Never - I/O tags are never read explicitly, only spontaneously received values are collected • Once - I/O tags are read explicitly only after the start (or restart) of the communication • Always - I/O tags are read explicitly in each reading cycle (according to the setting of time parameters of the station)	Never Once Always	Once
Read Group Priority	Reading priority used for explicit reading of I/O tags of type *Group address*: • System (0) • High (1) • Alarm (2) • Low (3)	-	Low
Read Group Timeout	Timeout for the response when explicitly reading I/O tags of type *Group address*. Allowed values are 0-60000 ms. A value of 0 means the default timeout defined by the KNX driver.	ms	0

I/O tag configuration

Possible I/O tag types: **TxtI, Di, Ai, Ci, TiR, TiA, TxtO, Dout, Ao, Co, ToR, ToA, Qi**



Individual components of the address:

Group address: group address (16-bit number). It has three possible formats:

- three-level address: main/middle/sub, where main = 0..31, middle = 0..7, sub = 0..255. Example: 1/2/24
- two-level address: main/sub, where main = 0..31, sub = 0..2047.
  Example: address 1/536 corresponds to 1/2/24 (because 2 * 256 + 24 = 536)
- direct number 1..65535 (group address 0 is not allowed).
  Example: address 2584 corresponds to 1/2/24 (because 1 * 2048 + 2 * 256 +24 = 2584)

Attribute: which attribute of the received message with a group value is published in the I/O tag:

- Value - the value itself. Its interpretation depends on the *Value type (reading)* setting
- Flags - text flags:
  - r - this is a value reading message
  - w - this is a value writing message
  - o - this is the answer
  - s - this is a secure communication message (secure)
- Source Address - KNX address of the device that sent the message (in the format area.line.device, eg 1.0.24)
- Priorities - numerical priority of the message:
  - System (0)
  - High (1)
  - Alarm (2)
  - Low (3)
- Size - the size of the received value in bits

Note: when explicitly reading I/O tags of the *Group address* type, only those that have *Attribute*=Value configured are read. In addition to the value, other attributes are extracted from the received response.

Note: when processing spontaneous messages with group values, the attributes Flags, Source Address, Priority, Size, and finally, Value are processed one after the other.

Value type (reading): for Attribute=Value it specifies a way of interpreting the received value:

- Unsigned Int - the value is interpreted as an unsigned integer (or as a True/False value)
- Signed Int - the value is interpreted as a signed integer
- Float - the value is interpreted as a real number (2, 4 or 8-byte)
- String - the value is interpreted as a string (the string in the KNX protocol has 14 characters)
- TimeOfDay (3B) - 3-byte value is interpreted as time of day (DPT_TimeOfDay) - value type must be TiR/ToR/Ai/Ao/Ci/Co
- Date (3B) - 3-byte value is interpreted as a date (DPT_Date) - value type must be TiA/ToA
- DateTime (8B) - 8-byte value is interpreted as a date and time (DPT_DateTime) - value type must be TiA/ToA

Note: for an I/O tag with a value of type TiA/ToA, it is possible to process a value of type *DPT_DateTime* (8-byte date and time) - it is necessary to set *Value type (reading) = Unsigned Int*

Value subtype (writing): for Attribute=Value, how to encode the value when writing:

- for Value type = *Unsigned Int*:
  - Bool
  - TwoBit
  - FourBit
  - SixBit
  - Unsigned Byte (1B)
  - Unsigned Short (2B)
  - Unsigned Int (4B)
  - Unsigned Long (8B)
- for Value type = *Signed Int*:
  - Signed Byte (1B)
  - Signed Short (2B)
  - Signed Int (4B)
  - Signed Long (8B)
- for Value type = *Float*:
  - Short Float (2B)
  - Float (4B)
  - Double (8B)
- for Value type = *String*:
  - String (14B)

Note: I/O tags that have the *Output mode* parameter set to *Command* in the *Output control* tab will not be read. In the KNX protocol, there are common objects that can be written to but not read from, so reading would end with an error that would invalidate the I/O tag value in D2000 - such I/O tag must be configured as a *Command*.

Browse

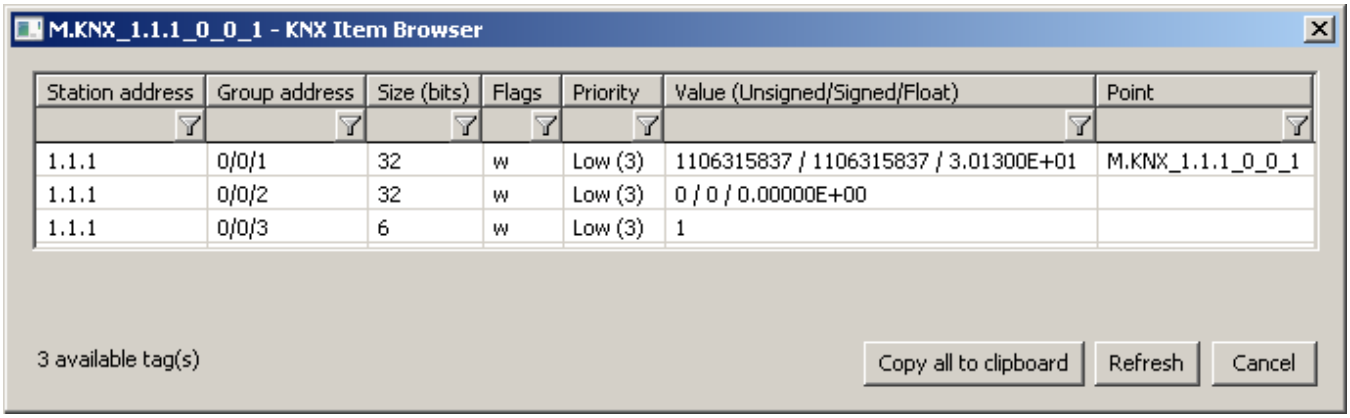
For the I/O tags, it is possible to discover the list of objects and their data types, as long as the KOM process is running and communication with an outstation is established.
Clicking the *Browse* button opens the *KNX Item Browser* window and displays a list of objects that have been read so far. The object list is created dynamically as a result of received messages (responses read requests as well as spontaneously arrived values).

The list of objects is dynamic, i.e. when a new value arrives in the KOM process, it is updated. Filtering in individual columns is also supported, asterisks can be used in the mask (eg 0/0/*).

Double-clicking on a particular line will cause the *Group address* parameter to be inserted into the configuration of the I/O tag from which the *KNX Item Browser* window was opened.

The Refresh button clears the list of values in both the CNF and the KOM process.

The *Value* column contains the received value interpreted as an unsigned integer, a signed integer (only if different from an unsigned integer), and a real number (if it is 16/32/64 bits long), or as text (if it is 14*8 bits long).



Literature

https://en.wikipedia.org/wiki/KNX_(standard)

Changes and modifications

-

Document revisions

- Ver. 1.0 - May 14th, 2020 - document creation.
- Ver. 1.1 - July 11th, 2022 - support for reading DPT_TimeOfDay, DPT_Date, DPT_DateTime.

Related pages:

Communication protocols