

libknx

Generated by Doxygen 1.8.8

Tue Mar 31 2015 02:12:04

Contents

1	KNX interface library	1
2	Namespace Index	3
2.1	Namespace List	3
3	Class Index	5
3.1	Class List	5
4	File Index	9
4.1	File List	9
5	Namespace Documentation	11
5.1	knx Namespace Reference	11
5.1.1	Detailed Description	13
6	Class Documentation	15
6.1	knx::config Class Reference	15
6.1.1	Detailed Description	15
6.2	knx::connection Class Reference	16
6.2.1	Detailed Description	16
6.2.2	Member Function Documentation	16
6.2.2.1	get	16
6.2.2.2	listen	16
6.2.2.3	set	17
6.2.2.4	start	17
6.2.2.5	stop	17
6.3	knx::dpt_6::data Class Reference	17
6.4	knx::dpt_21::data Class Reference	18
6.5	knx::dpt_26::data Class Reference	18
6.6	knx::dpt_12::data Class Reference	18
6.7	knx::dpt_219::data Class Reference	18
6.8	knx::dpt_13::data Class Reference	19
6.9	knx::dpt_232::data Class Reference	19

6.10	knx::dpt_7::data Class Reference	19
6.11	knx::dpt_14::data Class Reference	19
6.12	knx::dpt_15::data Class Reference	19
6.13	knx::dpt_4::data Class Reference	20
6.14	knx::dpt_8::data Class Reference	20
6.15	knx::dpt_16::data Class Reference	20
6.16	knx::dpt_9::data Class Reference	20
6.17	knx::dpt_17::data Class Reference	21
6.18	knx::dpt_18::data Class Reference	21
6.19	knx::dpt_5::data Class Reference	21
6.20	knx::dpt_19::data Class Reference	21
6.21	knx::data_point< dpt > Class Template Reference	22
6.22	knx::dpt_11::date Class Reference	22
6.22.1	Detailed Description	22
6.23	knx::dpt Class Reference	22
6.24	knx::dpt_1 Class Reference	23
6.24.1	Detailed Description	23
6.25	knx::dpt_10 Class Reference	23
6.25.1	Detailed Description	24
6.26	knx::dpt_10_001 Class Reference	24
6.26.1	Detailed Description	24
6.27	knx::dpt_11 Class Reference	24
6.27.1	Detailed Description	25
6.28	knx::dpt_11_001 Class Reference	25
6.28.1	Detailed Description	25
6.29	knx::dpt_12 Class Reference	25
6.29.1	Detailed Description	26
6.30	knx::dpt_12_001 Class Reference	26
6.30.1	Detailed Description	26
6.31	knx::dpt_13 Class Reference	26
6.31.1	Detailed Description	27
6.32	knx::dpt_13_001 Class Reference	27
6.32.1	Detailed Description	27
6.33	knx::dpt_14 Class Reference	28
6.33.1	Detailed Description	28
6.34	knx::dpt_14_000 Class Reference	28
6.34.1	Detailed Description	28
6.35	knx::dpt_15 Class Reference	29
6.35.1	Detailed Description	29
6.36	knx::dpt_15_000 Class Reference	29

6.36.1 Detailed Description	30
6.37 knx::dpt_16 Class Reference	30
6.37.1 Detailed Description	30
6.38 knx::dpt_16_000 Class Reference	30
6.38.1 Detailed Description	31
6.39 knx::dpt_17 Class Reference	31
6.39.1 Detailed Description	31
6.40 knx::dpt_17_001 Class Reference	31
6.40.1 Detailed Description	32
6.41 knx::dpt_18 Class Reference	32
6.41.1 Detailed Description	32
6.42 knx::dpt_18_001 Class Reference	32
6.42.1 Detailed Description	33
6.43 knx::dpt_19 Class Reference	33
6.43.1 Detailed Description	33
6.44 knx::dpt_19_001 Class Reference	34
6.44.1 Detailed Description	34
6.45 knx::dpt_1_001 Class Reference	34
6.45.1 Detailed Description	34
6.46 knx::dpt_2 Class Reference	35
6.46.1 Detailed Description	35
6.47 knx::dpt_20 Class Reference	35
6.47.1 Detailed Description	35
6.48 knx::dpt_20_011 Class Reference	35
6.48.1 Detailed Description	36
6.49 knx::dpt_21 Class Reference	36
6.49.1 Detailed Description	36
6.50 knx::dpt_219 Class Reference	37
6.50.1 Detailed Description	37
6.51 knx::dpt_219_001 Class Reference	37
6.51.1 Detailed Description	37
6.52 knx::dpt_21_001 Class Reference	38
6.52.1 Detailed Description	38
6.53 knx::dpt_232 Class Reference	38
6.53.1 Detailed Description	38
6.54 knx::dpt_232_600 Class Reference	39
6.54.1 Detailed Description	39
6.55 knx::dpt_26 Class Reference	39
6.55.1 Detailed Description	40
6.56 knx::dpt_26_001 Class Reference	40

6.56.1 Detailed Description	40
6.57 knx::dpt_2_001 Class Reference	40
6.57.1 Detailed Description	41
6.58 knx::dpt_3 Class Reference	41
6.58.1 Detailed Description	41
6.59 knx::dpt_3_007 Class Reference	41
6.59.1 Detailed Description	42
6.60 knx::dpt_4 Class Reference	42
6.60.1 Detailed Description	42
6.61 knx::dpt_4_001 Class Reference	42
6.61.1 Detailed Description	43
6.62 knx::dpt_5 Class Reference	43
6.62.1 Detailed Description	43
6.63 knx::dpt_5_001 Class Reference	43
6.63.1 Detailed Description	44
6.64 knx::dpt_6 Class Reference	44
6.64.1 Detailed Description	44
6.65 knx::dpt_6_001 Class Reference	45
6.65.1 Detailed Description	45
6.66 knx::dpt_7 Class Reference	45
6.66.1 Detailed Description	46
6.67 knx::dpt_7_001 Class Reference	46
6.67.1 Detailed Description	46
6.68 knx::dpt_8 Class Reference	46
6.68.1 Detailed Description	47
6.69 knx::dpt_8_001 Class Reference	47
6.69.1 Detailed Description	47
6.70 knx::dpt_9 Class Reference	47
6.70.1 Detailed Description	48
6.71 knx::dpt_9_001 Class Reference	48
6.71.1 Detailed Description	48
6.72 knx::group Class Reference	49
6.72.1 Detailed Description	49
6.72.2 Member Function Documentation	49
6.72.2.1 get_binary	49
6.72.2.2 get_name	49
6.72.2.3 set_by_binary_2	49
6.72.2.4 set_by_binary_3	50
6.72.2.5 set_by_name	50
6.72.2.6 set_by_number	50

6.72.2.7	set_by_number	50
6.73	knx::handle Class Reference	50
6.73.1	Detailed Description	51
6.74	knx::dpt_26::set_data Class Reference	51
6.75	knx::dpt_19::set_data Class Reference	51
6.75.1	Detailed Description	51
6.76	knx::dpt_18::set_data Class Reference	52
6.77	knx::dpt_21::set_data Class Reference	52
6.78	knx::dpt_15::set_data Class Reference	52
6.78.1	Detailed Description	52
6.79	knx::dpt_10::time Class Reference	53
6.79.1	Detailed Description	53
7	File Documentation	55
7.1	config.hpp File Reference	55
7.1.1	Detailed Description	55
7.2	connection.hpp File Reference	55
7.2.1	Detailed Description	56
7.3	data_point.hpp File Reference	56
7.3.1	Detailed Description	56
7.4	dpt.hpp File Reference	56
7.4.1	Detailed Description	60
7.4.2	Macro Definition Documentation	60
7.4.2.1	SUPPORTED_DPTS	60
7.5	group.hpp File Reference	60
7.5.1	Detailed Description	60
7.6	handle.hpp File Reference	61
7.6.1	Detailed Description	61
7.7	knx.hpp File Reference	61
7.7.1	Detailed Description	61
	Index	62

Chapter 1

KNX interface library

Author

Norbert Schmitz, knx@nagilo.de

Date

March 2015

Version

1.2.3

This library can be used to access the home automation bus system KNX using an IP gateway. More information on the bus may be found at www.knx.com.

Changes since version 1.2.2

- Added dpts. Now supporting: 1_001,2_001,3_007,4_001,5_001,6_001,7_001,8_001,9_001,10_001,11_001,12_001,13_001,14_000,15_000,16_000,17_001,18_001,19_001,20_011,21_001,26_001,219_001,232_600
- Added class `knx::group` for group address handling
- Supporting two part group addresses (at receive time 3 part addresses are assumed)

Changes since version 1.2.1

- Added support for dpt 7.001, 8.001
- Bugfix for dpt 9 & 10 (incorrect value setting)

Changes since version 1.0.0

- Added listener interface
- Added several new dpts. Now supporting dpt 1.001, 2.001, 3.007, 4.001, 5.001, 6.001, 9.001, 10.001.

The whole program including the header files are free to be used in any non-commercial application. A notification of usage to the author would be very nice.

Commercial use is strictly forbidden. In case you are interested in a commercial license please contact the author.

Example

A minimal code example would look like this when setting group 1/2/3 to on using a data point type of 1.001 .

```
#include "knx.hpp"
int main(int argc, char ** argv) {
    knx::config config(argc, argv);
    knx::connection connection(config);
    knx::handle handle(connection);
    connection.set<knx::dpt_switch>(knx::group("1/2/3"), knx::dpt_switch::ON);
    return 0;
}
```

Features

Although there are many eib/knx libraries available I started to develop an new knx library from scratch. The reasons for this are the following:

- Minimal dependencies:
Many existing libraries contain dependencies to additional libraries which make the compilation process harder. libknx depends solely on boost. No other dependencies are allowed.
- Platform independence:
libknx is portable. It is able to run on any linux operating system including Raspberry PI and on any Mac or Windows computer. The main development is realized on a Linux Debian 7.0 32bit machine but other distros and platforms will follow.
- Reduce to the maximum:
The library on its own does not contain any additional overhead. It is capable of writing and reading knx messages – nothing more and nothing less. Any additional ideas I or other may have should be based on libknx without extending the base system.

Current limitations

libknx is a brand new development which means that many features are currently missing. Most of them will be added in future. Version 1.2.3 has the following limitations:

- Only UDP communication is available

Ideas

The development of such a library directly creates tons of ideas what can be done with it. Besides others these are:

- Control your curtains using a Kinect
- Control your lights using a LeapMotion
- ...

Yours Norbert Schmitz

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

- [knx](#) Separated namespace to embed all libknx related classes 11

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

- [knx::config](#)
This class represents the configuration of the knx connection 15
- [knx::connection](#)
This class handles the ip connection(s) to the knx gateway 16
- [knx::dpt_6::data](#) 17
- [knx::dpt_21::data](#) 18
- [knx::dpt_26::data](#) 18
- [knx::dpt_12::data](#) 18
- [knx::dpt_219::data](#) 18
- [knx::dpt_13::data](#) 19
- [knx::dpt_232::data](#) 19
- [knx::dpt_7::data](#) 19
- [knx::dpt_14::data](#) 19
- [knx::dpt_15::data](#) 19
- [knx::dpt_4::data](#) 20
- [knx::dpt_8::data](#) 20
- [knx::dpt_16::data](#) 20
- [knx::dpt_9::data](#) 20
- [knx::dpt_17::data](#) 21
- [knx::dpt_18::data](#) 21
- [knx::dpt_5::data](#) 21
- [knx::dpt_19::data](#) 21
- [knx::data_point< dpt >](#) 22
- [knx::dpt_11::date](#)
Internal class storing a date value 22
- [knx::dpt](#) 22
- [knx::dpt_1](#)
Major class for all 1.XXX data point types 23
- [knx::dpt_10](#)
Major class for all 10.XXX data point types 23
- [knx::dpt_10_001](#)
Class holding a time value 24
- [knx::dpt_11](#)
Major class for all 11.XXX data point types 24
- [knx::dpt_11_001](#)
Class holding a date value 25
- [knx::dpt_12](#)
Major class for all 12.XXX data point types 25

knx::dpt_12_001	Class holding a 32 bit unsigned value	26
knx::dpt_13	Major class for all 13.XXX data point types	26
knx::dpt_13_001	Class holding a 32 bit signed value	27
knx::dpt_14	Major class for all 14.XXX data point types	28
knx::dpt_14_000	Class holding a 32 bit float value	28
knx::dpt_15	Major class for all 15.XXX data point types	29
knx::dpt_15_000	Class holding a access data type	29
knx::dpt_16	Major class for all 16.XXX data point types	30
knx::dpt_16_000	Class holding a fixed length string of up to 14 chars in ASCII	30
knx::dpt_17	Major class for all 17.XXX data point types	31
knx::dpt_17_001	Class holding a scene number	31
knx::dpt_18	Major class for all 18.XXX data point types	32
knx::dpt_18_001	Class holding a scene control value	32
knx::dpt_19	Major class for all 19.XXX data point types	33
knx::dpt_19_001	Class holding a date time value	34
knx::dpt_1_001	Data point type 1.001 simple boolean value	34
knx::dpt_2	Major class for all 2.XXX data point types	35
knx::dpt_20	Major class for all 20.XXX data point types	35
knx::dpt_20_011	Class holding a error class system value	35
knx::dpt_21	Major class for all 21.XXX data point types	36
knx::dpt_219	Major class for all 219.XXX data point types	37
knx::dpt_219_001	Class holding a alarm info	37
knx::dpt_21_001	Class holding a status gen value	38
knx::dpt_232	Major class for all 232.XXX data point types	38
knx::dpt_232_600	Class holding rgb color value	39
knx::dpt_26	Major class for all 26.XXX data point types	39
knx::dpt_26_001	Class holding a scene info value	40
knx::dpt_2_001	Data point type 2.001 control value	40
knx::dpt_3	Major class for all 3.XXX data point types	41

knx::dpt_3_007	Data point type 3.007 dimming control	41
knx::dpt_4	Major class for all 4.XXX data point types	42
knx::dpt_4_001	Data point type 4.001 char ascii	42
knx::dpt_5	Major class for all 5.XXX data point types	43
knx::dpt_5_001	Data point type 5.001 unsigned 8 bit	43
knx::dpt_6	Major class for all 6.XXX signed 8 bit value	44
knx::dpt_6_001	Data point type 6.001 percent v8	45
knx::dpt_7	Major class for all 7.XXX unsigned signed 16 bit value	45
knx::dpt_7_001	Data point type 7.001 unsigned 16bit pulse counter	46
knx::dpt_8	Major class for all 8.XXX signed 16 bit value	46
knx::dpt_8_001	Data point type 8.001 16bit signed pulse counter	47
knx::dpt_9	Major class for all 9.XXX data point types	47
knx::dpt_9_001	Data point type 9.001 value temp	48
knx::group	This class represents a KNX group address which might be either 3 compnent "a/b/c" or 2 component "a/b"	49
knx::handle	Thread management for connection handling	50
knx::dpt_26::set_data	51
knx::dpt_19::set_data	TODO	51
knx::dpt_18::set_data	52
knx::dpt_21::set_data	52
knx::dpt_15::set_data	TODO add documentation	52
knx::dpt_10::time	Internal class storing a time value	53

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

config.hpp	55
connection.hpp	55
data_point.hpp	56
dpt.hpp	56
group.hpp	60
handle.hpp	61
knx.hpp	61

Chapter 5

Namespace Documentation

5.1 knx Namespace Reference

separated namespace to embed all libknx related classes

Classes

- class [config](#)
This class represents the configuration of the knx connection.
- class [connection](#)
This class handles the ip connection(s) to the knx gateway.
- class [data_point](#)
- class [dpt](#)
- class [dpt_1](#)
major class for all 1.XXX data point types
- class [dpt_10](#)
major class for all 10.XXX data point types
- class [dpt_10_001](#)
class holding a time value
- class [dpt_11](#)
major class for all 11.XXX data point types
- class [dpt_11_001](#)
class holding a date value
- class [dpt_12](#)
major class for all 12.XXX data point types
- class [dpt_12_001](#)
class holding a 32 bit unsigned value
- class [dpt_13](#)
major class for all 13.XXX data point types
- class [dpt_13_001](#)
class holding a 32 bit signed value
- class [dpt_14](#)
major class for all 14.XXX data point types
- class [dpt_14_000](#)
class holding a 32 bit float value
- class [dpt_15](#)
major class for all 15.XXX data point types

- class [dpt_15_000](#)
class holding a access data type
- class [dpt_16](#)
major class for all 16.XXX data point types
- class [dpt_16_000](#)
class holding a fixed length string of up to 14 chars in ASCII
- class [dpt_17](#)
major class for all 17.XXX data point types
- class [dpt_17_001](#)
class holding a scene number
- class [dpt_18](#)
major class for all 18.XXX data point types
- class [dpt_18_001](#)
class holding a scene control value
- class [dpt_19](#)
major class for all 19.XXX data point types
- class [dpt_19_001](#)
class holding a date time value
- class [dpt_1_001](#)
data point type 1.001 simple boolean value
- class [dpt_2](#)
major class for all 2.XXX data point types
- class [dpt_20](#)
major class for all 20.XXX data point types
- class [dpt_20_011](#)
class holding a error class system value
- class [dpt_21](#)
major class for all 21.XXX data point types
- class [dpt_219](#)
major class for all 219.XXX data point types
- class [dpt_219_001](#)
class holding a alarm info
- class [dpt_21_001](#)
class holding a status gen value
- class [dpt_232](#)
major class for all 232.XXX data point types
- class [dpt_232_600](#)
class holding rgb color value
- class [dpt_26](#)
major class for all 26.XXX data point types
- class [dpt_26_001](#)
class holding a scene info value
- class [dpt_2_001](#)
data point type 2.001 control value
- class [dpt_3](#)
major class for all 3.XXX data point types
- class [dpt_3_007](#)
data point type 3.007 dimming control
- class [dpt_4](#)
major class for all 4.XXX data point types
- class [dpt_4_001](#)

- data point type 4.001 char ascii*
- class [dpt_5](#)
 - major class for all 5.XXX data point types*
- class [dpt_5_001](#)
 - data point type 5.001 unsigned 8 bit*
- class [dpt_6](#)
 - major class for all 6.XXX signed 8 bit value*
- class [dpt_6_001](#)
 - data point type 6.001 percent v8*
- class [dpt_7](#)
 - major class for all 7.XXX unsigned signed 16 bit value*
- class [dpt_7_001](#)
 - data point type 7.001 unsigned 16bit pulse counter*
- class [dpt_8](#)
 - major class for all 8.XXX signed 16 bit value*
- class [dpt_8_001](#)
 - data point type 8.001 16bit signed pulse counter*
- class [dpt_9](#)
 - major class for all 9.XXX data point types*
- class [dpt_9_001](#)
 - data point type 9.001 value temp*
- class [group](#)
 - This class represents a KNX group address which might be either 3 compnent "a/b/c" or 2 component "a/b".*
- class [handle](#)
 - Thread management for connection handling.*

Typedefs

- typedef [dpt_1_001](#) [dpt_switch](#)
 - data point type 1.001 is better known as switch*
- typedef [dpt_5_001](#) [dpt_scaling](#)
 - 5.001 is better known as scaling value*
- typedef [dpt_10_001](#) [dpt_timeofday](#)
 - 10.001 is known as time of day value including hours. minutes, seconds and day of the week*
- typedef `boost::mpl::vector<>` [dpt_types](#)

Enumerations

- enum [dpt_index](#)

Functions

- `std::ostream & operator<< (std::ostream &os, const dpt_10::time &time)`
- `template<class dpt >
KNX_IMPORT_EXPORT std::ostream & operator<< (std::ostream &os, const data_point< dpt > &data_point)`

5.1.1 Detailed Description

separated namespace to embed all libknx related classes

Chapter 6

Class Documentation

6.1 knx::config Class Reference

This class represents the configuration of the knx connection.

```
#include <config.hpp>
```

Public Member Functions

- [config](#) (int argc, char *argv[])
constructor taking all command line arguments
- [~config](#) ()
empty destructor

Public Attributes

- [std::string local_control_host](#)
name or ip address of the local host (control connection)
- [int local_control_port](#)
communication port for control messages
- [std::string local_data_host](#)
name or ip address of the local host (data connection)
- [int local_data_port](#)
communication port for data messages
- [std::string remote_host](#)
name or ip address of the knx gateway
- [int remote_port](#)
port of the gateway (default 3671)
- [std::string logging_filename](#)
name of the logging file
- [bool logging_activated](#)
indication if logging is active

6.1.1 Detailed Description

This class represents the configuration of the knx connection.

The documentation for this class was generated from the following file:

- [config.hpp](#)

6.2 knx::connection Class Reference

This class handles the ip connection(s) to the knx gateway.

```
#include <connection.hpp>
```

Public Member Functions

- `connection` (const `knx::config` &`config`)
creates the connection with given config
- `~connection` ()
simple destructor
- void `start` ()
blocking start of the connection background thread
- void `stop` ()
non-blocking stop of the background thread
- template<typename `data_type` >
void `set` (`knx::group group`, typename `data_type::major_type::set_type data`)
central function to set any knx group value on the bus
- template<typename `data_type` >
bool `get` (`knx::group group`, typename `data_type::major_type::set_type &data`)
central function to get any knx group value from the bus
- template<typename `data_type` >
bool `listen` (`knx::group group`, boost::function< void(typename `data_type::major_type::set_type` &) > `callback`)
central function to continuously get any knx group value from the bus

6.2.1 Detailed Description

This class handles the ip connection(s) to the knx gateway.

6.2.2 Member Function Documentation

6.2.2.1 template<typename `data_type` > bool `knx::connection::get` (`knx::group group`, typename `data_type::major_type::set_type & data`)

central function to get any knx group value from the bus

This function is used to get any value from the knx bus.

Parameters

<i>group</i>	A valid group id as string (e.g. "1/2/3")
--------------	---

Returns

true if read was successful – false otherwise

6.2.2.2 template<typename `data_type` > bool `knx::connection::listen` (`knx::group group`, boost::function< void(typename `data_type::major_type::set_type` &) > `callback`)

central function to continuously get any knx group value from the bus

This function is used to listen for any value from the knx bus.

A minimal example would be:


```

#include "knx.hpp"
#include "debug.hpp"
void my_callback ( knx::dpt_1_001::set_type & data )
{
    std::cout << "listened to " << data << std::endl;
}
int main ( int argc, char ** argv )
{
    knx::config config ( argc, argv );
    knx::connection connection ( config );
    knx::handle handle ( connection );
    boost::this_thread::sleep ( boost::posix_time::seconds ( 2 ) );
    connection.listen<knx::dpt_1_001> ( "0/0/1", my_callback );
    boost::this_thread::sleep( boost::posix_time::seconds ( 20 ) );
    return 0;
}

```

Parameters

<i>group</i>	A valid group id as string (e.g. "1/2/3")
<i>callback</i>	A function to be called whenever data has been received

Returns

true if registering was successful – false otherwise

6.2.2.3 `template<typename data_type > void knx::connection::set (knx::group group, typename data_type::major_type::set_type data)`

central function to set any knx group value on the bus

This function is used to set any value on the knx bus.

6.2.2.4 `void knx::connection::start ()`

blocking start of the connection background thread

This function is a blocking call to start the background thread. It is normally only called by the handler.

Warning

you should not call this function from your code.

6.2.2.5 `void knx::connection::stop ()`

non-blocking stop of the background thread

This function send the termination signals and waits for the thread to stop.

The documentation for this class was generated from the following file:

- [connection.hpp](#)

6.3 knx::dpt_6::data Class Reference

Public Member Functions

- **data** (int8_t real_signed_value=0)

Public Attributes

- `int8_t real_signed_value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.4 knx::dpt_21::data Class Reference

Public Attributes

- `uint8_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.5 knx::dpt_26::data Class Reference

Public Attributes

- `uint8_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.6 knx::dpt_12::data Class Reference

Public Attributes

- `uint32_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.7 knx::dpt_219::data Class Reference

Public Attributes

- `uint8_t log_number`
- `uint8_t alarm_priority`
- `uint8_t application_area`
- `uint8_t error_class`
- `uint8_t attributes`
- `uint8_t alarm_status_attributes`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.8 knx::dpt_13::data Class Reference

Public Attributes

- `int32_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.9 knx::dpt_232::data Class Reference

Public Attributes

- `uint8_t r`
- `uint8_t g`
- `uint8_t b`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.10 knx::dpt_7::data Class Reference

Public Attributes

- `uint16_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.11 knx::dpt_14::data Class Reference

Public Attributes

- `float value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.12 knx::dpt_15::data Class Reference

Public Attributes

- `uint8_t d65`
- `uint8_t d43`
- `uint8_t d21`

- `uint8_t epdc_index`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.13 knx::dpt_4::data Class Reference

Public Member Functions

- **data** (uint8_t character=0)

Public Attributes

- `uint8_t character`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.14 knx::dpt_8::data Class Reference

Public Attributes

- `int16_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.15 knx::dpt_16::data Class Reference

Public Attributes

- `char text [14]`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.16 knx::dpt_9::data Class Reference

Public Member Functions

- **data** (float float_value=0.f)

Public Attributes

- `uint16_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.17 knx::dpt_17::data Class Reference

Public Attributes

- `uint8_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.18 knx::dpt_18::data Class Reference

Public Attributes

- `uint8_t value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.19 knx::dpt_5::data Class Reference

Public Member Functions

- `data (uint8_t unsigned_value=0)`

Public Attributes

- `uint8_t unsigned_value`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.20 knx::dpt_19::data Class Reference

Public Attributes

- `uint32_t date`
- `uint32_t time`

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.21 knx::data_point< dpt > Class Template Reference

Public Types

- typedef dpt::major_type **major_type**
- typedef major_type::set_type **set_type**
- typedef major_type::data_type **data_type**

Public Member Functions

- **data_point** (set_type value)
- std::string **describe** () const
- set_type **get** () const
- void **set** (set_type value)

The documentation for this class was generated from the following file:

- [data_point.hpp](#)

6.22 knx::dpt_11::date Class Reference

internal class storing a date value

```
#include <dpt.hpp>
```

Public Member Functions

- bool **operator==** (const [date](#) &other)
- [date](#) ()
creates a current date value

Public Attributes

- uint8_t **day**
- uint8_t **month**
- uint8_t **year**

6.22.1 Detailed Description

internal class storing a date value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.23 knx::dpt Class Reference

Static Public Member Functions

- template<typename data_type >
static std::string **describe** (const typename data_type::major_type::set_type &data)

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.24 knx::dpt_1 Class Reference

major class for all 1.XXX data point types

```
#include <dpt.hpp>
```

Public Types

- typedef uint8_t **data_type**
- typedef bool **set_type**

Static Public Member Functions

- static void **set** (set_type value, data_type &data)
- static set_type **get** (const data_type &data)

6.24.1 Detailed Description

major class for all 1.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.25 knx::dpt_10 Class Reference

major class for all 10.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [time](#)
internal class storing a time value

Public Types

- typedef uint32_t **data_type**
- typedef [time](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, data_type &data)
- static [set_type](#) **get** (const data_type &data)

6.25.1 Detailed Description

major class for all 10.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.26 knx::dpt_10_001 Class Reference

class holding a time value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_10](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)
- static [major_type::time](#) **now** ()

Static Public Attributes

- static const int **INDEX** = DPT_10_001

6.26.1 Detailed Description

class holding a time value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.27 knx::dpt_11 Class Reference

major class for all 11.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [date](#)
internal class storing a date value

Public Types

- typedef uint32_t **data_type**
- typedef [date](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, data_type &data)
- static [set_type](#) **get** (const data_type &data)

6.27.1 Detailed Description

major class for all 11.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.28 knx::dpt_11_001 Class Reference

class holding a date value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_11](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_11_001

6.28.1 Detailed Description

class holding a date value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.29 knx::dpt_12 Class Reference

major class for all 12.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef `data` **data_type**
- typedef `uint32_t` **set_type**

Static Public Member Functions

- static void **set** (set_type value, data_type &data)
- static set_type **get** (const data_type &data)

6.29.1 Detailed Description

major class for all 12.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.30 knx::dpt_12_001 Class Reference

class holding a 32 bit unsigned value

```
#include <dpt.hpp>
```

Public Types

- typedef `dpt_12` **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_12_001

6.30.1 Detailed Description

class holding a 32 bit unsigned value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.31 knx::dpt_13 Class Reference

major class for all 13.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef uint32_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.31.1 Detailed Description

major class for all 13.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.32 knx::dpt_13_001 Class Reference

class holding a 32 bit signed value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_13](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_13_001

6.32.1 Detailed Description

class holding a 32 bit signed value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.33 knx::dpt_14 Class Reference

major class for all 14.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef uint32_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.33.1 Detailed Description

major class for all 14.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.34 knx::dpt_14_000 Class Reference

class holding a 32 bit float value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_14](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_14_000

6.34.1 Detailed Description

class holding a 32 bit float value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.35 knx::dpt_15 Class Reference

major class for all 15.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)
- class [set_data](#)

TODO add documentation.

Public Types

- typedef [data](#) **data_type**
- typedef [set_data](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, [data_type](#) &data)
- static [set_type](#) **get** (const [data_type](#) &data)

6.35.1 Detailed Description

major class for all 15.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.36 knx::dpt_15_000 Class Reference

class holding a access data type

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_15](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_15_000

6.36.1 Detailed Description

class holding a access data type

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.37 knx::dpt_16 Class Reference

major class for all 16.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef std::string **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.37.1 Detailed Description

major class for all 16.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.38 knx::dpt_16_000 Class Reference

class holding a fixed length string of up to 14 chars in ASCII

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_16](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_16_000

6.38.1 Detailed Description

class holding a fixed length string of up to 14 chars in ASCII

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.39 knx::dpt_17 Class Reference

major class for all 17.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef uint8_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.39.1 Detailed Description

major class for all 17.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.40 knx::dpt_17_001 Class Reference

class holding a scene number

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_17](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_17_001

6.40.1 Detailed Description

class holding a scene number

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.41 knx::dpt_18 Class Reference

major class for all 18.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)
- class [set_data](#)

Public Types

- typedef [data](#) **data_type**
- typedef [set_data](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, [data_type](#) &data)
- static [set_type](#) **get** (const [data_type](#) &data)

6.41.1 Detailed Description

major class for all 18.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.42 knx::dpt_18_001 Class Reference

class holding a scene control value

```
#include <dpt.hpp>
```


Public Types

- typedef [dpt_18](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_18_001

6.42.1 Detailed Description

class holding a scene control value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.43 knx::dpt_19 Class Reference

major class for all 19.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)
- class [set_data](#)

TODO.

Public Types

- typedef [data](#) **data_type**
- typedef [set_data](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, [data_type](#) &data)
- static [set_type](#) **get** (const [data_type](#) &data)

6.43.1 Detailed Description

major class for all 19.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.44 knx::dpt_19_001 Class Reference

class holding a date time value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_19](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_19_001

6.44.1 Detailed Description

class holding a date time value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.45 knx::dpt_1_001 Class Reference

data point type 1.001 simple boolean value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_1](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_1_001
- static const [major_type::set_type](#) **ON**
- static const [major_type::set_type](#) **OFF**

6.45.1 Detailed Description

data point type 1.001 simple boolean value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.46 knx::dpt_2 Class Reference

major class for all 2.XXX data point types

```
#include <dpt.hpp>
```

Public Types

- typedef uint8_t **data_type**
- typedef std::pair< bool, bool > **set_type**

Static Public Member Functions

- static void **set** (set_type value, data_type &data)
- static set_type **get** (const data_type &data)

6.46.1 Detailed Description

major class for all 2.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.47 knx::dpt_20 Class Reference

major class for all 20.XXX data point types

```
#include <dpt.hpp>
```

Public Types

- typedef uint16_t **data_type**
- typedef uint8_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, data_type &data)
- static set_type **get** (const data_type &data)

6.47.1 Detailed Description

major class for all 20.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.48 knx::dpt_20_011 Class Reference

class holding a error class system value

```
#include <dpt.hpp>
```

Public Types

- enum {
 NO_FAULT, **GENERAL_DEVICE_FAULT**, **COMMUNICATION_FAULT**, **CONFIGURATION_FAULT**,
 HARDWARE_FAULT, **SOFTWARE_FAULT** }
- typedef [dpt_20](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_20_011

6.48.1 Detailed Description

class holding a error class system value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.49 knx::dpt_21 Class Reference

major class for all 21.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)
- class [set_data](#)

Public Types

- typedef [data](#) **data_type**
- typedef [set_data](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, [data_type](#) &data)
- static [set_type](#) **get** (const [data_type](#) &data)

6.49.1 Detailed Description

major class for all 21.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.50 knx::dpt_219 Class Reference

major class for all 219.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef [data](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, [data_type](#) &data)
- static [set_type](#) **get** (const [data_type](#) &data)

6.50.1 Detailed Description

major class for all 219.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.51 knx::dpt_219_001 Class Reference

class holding a alarm info

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_219](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_219_001

6.51.1 Detailed Description

class holding a alarm info

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.52 knx::dpt_21_001 Class Reference

class holding a status gen value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_21](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_21_001

6.52.1 Detailed Description

class holding a status gen value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.53 knx::dpt_232 Class Reference

major class for all 232.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef [data](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, [data_type](#) &data)
- static [set_type](#) **get** (const [data_type](#) &data)

6.53.1 Detailed Description

major class for all 232.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.54 knx::dpt_232_600 Class Reference

class holding rgb color value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_232](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_232_600

6.54.1 Detailed Description

class holding rgb color value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.55 knx::dpt_26 Class Reference

major class for all 26.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)
- class [set_data](#)

Public Types

- typedef [data](#) **data_type**
- typedef [set_data](#) **set_type**

Static Public Member Functions

- static void **set** ([set_type](#) value, [data_type](#) &data)
- static [set_type](#) **get** (const [data_type](#) &data)

6.55.1 Detailed Description

major class for all 26.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.56 knx::dpt_26_001 Class Reference

class holding a scene info value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_26](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_26_001

6.56.1 Detailed Description

class holding a scene info value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.57 knx::dpt_2_001 Class Reference

data point type 2.001 control value

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_2](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const [major_type::set_type](#) &data)

Static Public Attributes

- static const int **INDEX** = DPT_2_001
- static const major_type::set_type **NO_CONTROL**
- static const major_type::set_type **CONTROL_VALUE_ZERO**
- static const major_type::set_type **CONTROL_VALUE_ONE**

6.57.1 Detailed Description

data point type 2.001 control value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.58 knx::dpt_3 Class Reference

major class for all 3.XXX data point types

```
#include <dpt.hpp>
```

Public Types

- typedef uint8_t **data_type**
- typedef std::pair< bool, uint8_t > **set_type**

Static Public Member Functions

- static void **set** (set_type value, data_type &data)
- static set_type **get** (const data_type &data)

6.58.1 Detailed Description

major class for all 3.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.59 knx::dpt_3_007 Class Reference

data point type 3.007 dimming control

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_3](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_3_007

6.59.1 Detailed Description

data point type 3.007 dimming control

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.60 knx::dpt_4 Class Reference

major class for all 4.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef uint8_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.60.1 Detailed Description

major class for all 4.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.61 knx::dpt_4_001 Class Reference

data point type 4.001 char ascii

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_4](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_4_001

6.61.1 Detailed Description

data point type 4.001 char ascii

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.62 knx::dpt_5 Class Reference

major class for all 5.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef uint8_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.62.1 Detailed Description

major class for all 5.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.63 knx::dpt_5_001 Class Reference

data point type 5.001 unsigned 8 bit

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_5](#) **major_type**
- typedef TYPENAME
major_type::set_type **set_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_5_001

6.63.1 Detailed Description

data point type 5.001 unsigned 8 bit

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.64 knx::dpt_6 Class Reference

major class for all 6.XXX signed 8 bit value

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef int8_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.64.1 Detailed Description

major class for all 6.XXX signed 8 bit value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.65 knx::dpt_6_001 Class Reference

data point type 6.001 percent v8

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_6](#) **major_type**
- typedef TYPENAME
major_type::set_type **set_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_6_001

6.65.1 Detailed Description

data point type 6.001 percent v8

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.66 knx::dpt_7 Class Reference

major class for all 7.XXX unsigned signed 16 bit value

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef uint16_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

6.66.1 Detailed Description

major class for all 7.XXX unsigned signed 16 bit value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.67 knx::dpt_7_001 Class Reference

data point type 7.001 unsigned 16bit pulse counter

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_7](#) **major_type**
- typedef TYPENAME
major_type::set_type **set_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_7_001

6.67.1 Detailed Description

data point type 7.001 unsigned 16bit pulse counter

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.68 knx::dpt_8 Class Reference

major class for all 8.XXX signed 16 bit value

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef int16_t **set_type**

Static Public Member Functions

- static void **set** (set_type value, data_type &data)
- static set_type **get** (const data_type &data)

6.68.1 Detailed Description

major class for all 8.XXX signed 16 bit value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.69 knx::dpt_8_001 Class Reference

data point type 8.001 16bit signed pulse counter

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_8](#) **major_type**
- typedef TYPENAME
major_type::set_type **set_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_8_001

6.69.1 Detailed Description

data point type 8.001 16bit signed pulse counter

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.70 knx::dpt_9 Class Reference

major class for all 9.XXX data point types

```
#include <dpt.hpp>
```

Classes

- class [data](#)

Public Types

- typedef [data](#) **data_type**
- typedef float **set_type**

Static Public Member Functions

- static void **set** (set_type value, [data_type](#) &data)
- static set_type **get** (const [data_type](#) &data)

Static Public Attributes

- static const uint16_t **INVALID_DATA** = 0x7fff
- static const uint16_t **MAX** = 0x7ffe
- static const uint16_t **MIN** = 0xf800
- static const float **MAX_FLOAT** = 670760.96f
- static const float **MIN_FLOAT** = -671088.64f

6.70.1 Detailed Description

major class for all 9.XXX data point types

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.71 knx::dpt_9_001 Class Reference

data point type 9.001 value temp

```
#include <dpt.hpp>
```

Public Types

- typedef [dpt_9](#) **major_type**

Static Public Member Functions

- static std::string **describe** (const major_type::set_type &data)

Static Public Attributes

- static const int **INDEX** = DPT_9_001

6.71.1 Detailed Description

data point type 9.001 value temp

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.72 knx::group Class Reference

This class represents a KNX group address which might be either 3 component "a/b/c" or 2 component "a/b".

```
#include <group.hpp>
```

Public Member Functions

- [group](#) ()
Sets group to 0/0/0 by default.
- [group](#) (std::string name)
Assigns a group address by name. Sets default 0/0/0 if string is invalid. 2 component addresses may be used.
- [group](#) (uint8_t a, uint8_t b, uint8_t c)
Binary assign a 3 component group address.
- [group](#) (uint8_t a, uint8_t b)
Binary assign a 2 component group address.
- std::string [get_name](#) ()
- uint16_t [get_binary](#) ()
- void [set_by_number](#) (uint16_t a, uint16_t b, uint16_t c)
- void [set_by_number](#) (uint16_t a, uint16_t b)
- void [set_by_name](#) (std::string name)
- void [set_by_binary_2](#) (uint16_t binary)
- void [set_by_binary_3](#) (uint16_t binary)
- bool [operator==](#) (knx::group other)
Comparison operator for rapid equality checks.

6.72.1 Detailed Description

This class represents a KNX group address which might be either 3 component "a/b/c" or 2 component "a/b".

6.72.2 Member Function Documentation

6.72.2.1 uint16_t knx::group::get_binary ()

Get the binary group address value as 16 bit int.

Returns

Binary representation of the group address.

6.72.2.2 std::string knx::group::get_name ()

Gives the current set address as string.

Returns

The group address as string.

6.72.2.3 void knx::group::set_by_binary_2 (uint16_t binary)

Sets the whole address from binary value guessing a 2 component address in the form "a/b".

Parameters

<i>binary</i>	The address as 16 bit uint.
---------------	-----------------------------

6.72.2.4 void knx::group::set_by_binary_3 (uint16_t *binary*)

Sets the whole address from binary value guessing a 3 component address in the form "a/b/c".

Parameters

<i>binary</i>	The address as 16 bit uint.
---------------	-----------------------------

6.72.2.5 void knx::group::set_by_name (std::string *name*)

Set the address by name either as "1/2" or "1/2/3".

6.72.2.6 void knx::group::set_by_number (uint16_t *a*, uint16_t *b*, uint16_t *c*)

Set the address by a set of components in the form "a/b/c". If any value exceeds the range higher bits are ignored.

Parameters

<i>a</i>	First component 5 bits (0-31).
<i>b</i>	Second component 3 bits (0-7).
<i>c</i>	Third component 8 bits (0-255).

6.72.2.7 void knx::group::set_by_number (uint16_t *a*, uint16_t *b*)

Set the address by a set of components in the form "a/b". If any value exceeds the range higher bits are ignored.

Parameters

<i>a</i>	First component 5 bits (0-31).
<i>b</i>	Second component 11 bits (0-2047).

The documentation for this class was generated from the following file:

- [group.hpp](#)

6.73 knx::handle Class Reference

Thread management for connection handling.

```
#include <handle.hpp>
```

Public Member Functions

- [handle](#) (knx::connection &connection)
Creates new background thread.
- [~handle](#) ()
Stops and joins the thread.

6.73.1 Detailed Description

Thread management for connection handling.

This class handles all required background threads for asynchronous knx connection handling.

The documentation for this class was generated from the following file:

- [handle.hpp](#)

6.74 knx::dpt_26::set_data Class Reference

Public Attributes

- uint8_t **scene_number**
- bool **b**

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.75 knx::dpt_19::set_data Class Reference

TODO.

```
#include <dpt.hpp>
```

Public Attributes

- uint8_t **year**
- uint8_t **month**
- uint8_t **day_of_month**
- uint8_t **day_of_week**
- uint8_t **hour_of_day**
- uint8_t **minutes**
- uint8_t **seconds**
- bool **f**
- bool **wd**
- bool **nwd**
- bool **ny**
- bool **nd**
- bool **ndow**
- bool **nt**
- bool **suti**
- bool **clq**

6.75.1 Detailed Description

TODO.

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.76 knx::dpt_18::set_data Class Reference

Public Attributes

- bool **c**
false corresponds to activate and true to learn the scene number
- uint8_t **scene_number**

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.77 knx::dpt_21::set_data Class Reference

Public Attributes

- bool **out_of_service**
- bool **fault**
- bool **overridden**
- bool **in_alarm**
- bool **alarm_un_ack**

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.78 knx::dpt_15::set_data Class Reference

TODO add documentation.

```
#include <dpt.hpp>
```

Public Attributes

- uint8_t **d6**
- uint8_t **d5**
- uint8_t **d4**
- uint8_t **d3**
- uint8_t **d2**
- uint8_t **d1**
- bool **e**
- bool **p**
- bool **d**
- bool **c**
- uint8_t **index**

6.78.1 Detailed Description

TODO add documentation.

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

6.79 knx::dpt_10::time Class Reference

internal class storing a time value

```
#include <dpt.hpp>
```

Public Member Functions

- bool **operator==** (const [time](#) &other)
- [time](#) ()
creates a current time value

Public Attributes

- uint8_t **day**
- uint8_t **hour**
- uint8_t **minutes**
- uint8_t **seconds**

Static Public Attributes

- static const uint8_t **NODAY** = 0
- static const uint8_t **MONDAY** = 1
- static const uint8_t **TUESDAY** = 2
- static const uint8_t **WEDNESDAY** = 3
- static const uint8_t **THURSDAY** = 4
- static const uint8_t **FRIDAY** = 5
- static const uint8_t **SATURDAY** = 6
- static const uint8_t **SUNDAY** = 7

Friends

- std::ostream & **operator<<** (std::ostream &os, const [time](#) &[time](#))

6.79.1 Detailed Description

internal class storing a time value

The documentation for this class was generated from the following file:

- [dpt.hpp](#)

Chapter 7

File Documentation

7.1 config.hpp File Reference

```
#include "debug.hpp"  
#include <string>
```

Classes

- class [knx::config](#)

This class represents the configuration of the knx connection.

Namespaces

- [knx](#)

separated namespace to embed all libknx related classes

7.1.1 Detailed Description

This class holds the current configuration of the knx connection.

7.2 connection.hpp File Reference

```
#include "debug.hpp"  
#include <boost/function.hpp>  
#include "config.hpp"  
#include "dpt.hpp"  
#include "group.hpp"
```

Classes

- class [knx::connection](#)

This class handles the ip connection(s) to the knx gateway.

Namespaces

- [knx](#)
separated namespace to embed all libknx related classes

7.2.1 Detailed Description

User interface class for setting, getting and listening.

7.3 data_point.hpp File Reference

```
#include "debug.hpp"
#include "dpt.hpp"
#include <string>
```

Classes

- class [knx::data_point< dpt >](#)

Namespaces

- [knx](#)
separated namespace to embed all libknx related classes

Functions

- `template<class dpt >`
`KNX_IMPORT_EXPORT std::ostream & knx::operator<< (std::ostream &os, const data_point< dpt >`
`&data_point)`

7.3.1 Detailed Description

Helper class not intended for direct user access.

7.4 dpt.hpp File Reference

```
#include "debug.hpp"
#include <stdint.h>
#include <sstream>
#include <boost/date_time.hpp>
#include <boost/mpi/vector.hpp>
```

Classes

- class [knx::dpt](#)
- class [knx::dpt_1](#)
major class for all 1.XXX data point types

- class [knx::dpt_1_001](#)
data point type 1.001 simple boolean value
- class [knx::dpt_2](#)
major class for all 2.XXX data point types
- class [knx::dpt_2_001](#)
data point type 2.001 control value
- class [knx::dpt_3](#)
major class for all 3.XXX data point types
- class [knx::dpt_3_007](#)
data point type 3.007 dimming control
- class [knx::dpt_4](#)
major class for all 4.XXX data point types
- class [knx::dpt_4::data](#)
- class [knx::dpt_4_001](#)
data point type 4.001 char ascii
- class [knx::dpt_5](#)
major class for all 5.XXX data point types
- class [knx::dpt_5::data](#)
- class [knx::dpt_5_001](#)
data point type 5.001 unsigned 8 bit
- class [knx::dpt_6](#)
major class for all 6.XXX signed 8 bit value
- class [knx::dpt_6::data](#)
- class [knx::dpt_6_001](#)
data point type 6.001 percent v8
- class [knx::dpt_7](#)
major class for all 7.XXX unsigned signed 16 bit value
- class [knx::dpt_7::data](#)
- class [knx::dpt_7_001](#)
data point type 7.001 unsigned 16bit pulse counter
- class [knx::dpt_8](#)
major class for all 8.XXX signed 16 bit value
- class [knx::dpt_8::data](#)
- class [knx::dpt_8_001](#)
data point type 8.001 16bit signed pulse counter
- class [knx::dpt_9](#)
major class for all 9.XXX data point types
- class [knx::dpt_9::data](#)
- class [knx::dpt_9_001](#)
data point type 9.001 value temp
- class [knx::dpt_10](#)
major class for all 10.XXX data point types
- class [knx::dpt_10::time](#)
internal class storing a time value
- class [knx::dpt_10_001](#)
class holding a time value
- class [knx::dpt_11](#)
major class for all 11.XXX data point types
- class [knx::dpt_11::date](#)
internal class storing a date value
- class [knx::dpt_11_001](#)

- class holding a date value*
- class [knx::dpt_12](#)
 - major class for all 12.XXX data point types*
- class [knx::dpt_12::data](#)
- class [knx::dpt_12_001](#)
 - class holding a 32 bit unsigned value*
- class [knx::dpt_13](#)
 - major class for all 13.XXX data point types*
- class [knx::dpt_13::data](#)
- class [knx::dpt_13_001](#)
 - class holding a 32 bit signed value*
- class [knx::dpt_14](#)
 - major class for all 14.XXX data point types*
- class [knx::dpt_14::data](#)
- class [knx::dpt_14_000](#)
 - class holding a 32 bit float value*
- class [knx::dpt_15](#)
 - major class for all 15.XXX data point types*
- class [knx::dpt_15::data](#)
- class [knx::dpt_15::set_data](#)
 - TODO add documentation.*
- class [knx::dpt_15_000](#)
 - class holding a access data type*
- class [knx::dpt_16](#)
 - major class for all 16.XXX data point types*
- class [knx::dpt_16::data](#)
- class [knx::dpt_16_000](#)
 - class holding a fixed length string of up to 14 chars in ASCII*
- class [knx::dpt_17](#)
 - major class for all 17.XXX data point types*
- class [knx::dpt_17::data](#)
- class [knx::dpt_17_001](#)
 - class holding a scene number*
- class [knx::dpt_18](#)
 - major class for all 18.XXX data point types*
- class [knx::dpt_18::data](#)
- class [knx::dpt_18::set_data](#)
- class [knx::dpt_18_001](#)
 - class holding a scene control value*
- class [knx::dpt_19](#)
 - major class for all 19.XXX data point types*
- class [knx::dpt_19::data](#)
- class [knx::dpt_19::set_data](#)
 - TODO.*
- class [knx::dpt_19_001](#)
 - class holding a date time value*
- class [knx::dpt_20](#)
 - major class for all 20.XXX data point types*
- class [knx::dpt_20_011](#)
 - class holding a error class system value*
- class [knx::dpt_21](#)

major class for all 21.XXX data point types

- class [knx::dpt_21::data](#)
- class [knx::dpt_21::set_data](#)
- class [knx::dpt_21_001](#)

class holding a status gen value

- class [knx::dpt_26](#)

major class for all 26.XXX data point types

- class [knx::dpt_26::data](#)
- class [knx::dpt_26::set_data](#)
- class [knx::dpt_26_001](#)

class holding a scene info value

- class [knx::dpt_219](#)

major class for all 219.XXX data point types

- class [knx::dpt_219::data](#)
- class [knx::dpt_219_001](#)

class holding a alarm info

- class [knx::dpt_232](#)

major class for all 232.XXX data point types

- class [knx::dpt_232::data](#)
- class [knx::dpt_232_600](#)

class holding rgb color value

Namespaces

- [knx](#)

separated namespace to embed all libknx related classes

Macros

- #define **SUPPORTED_DPTS**
- #define **OP**(s, data, elem) BOOST_PP_CAT(DPT_, elem)
- #define **OP2**(s, data, elem) BOOST_PP_CAT(knx::dpt_, elem)
- #define **SEQ** BOOST_PP_SEQ_TRANSFORM(OP2, _, SUPPORTED_DPTS)
- #define **OP3**(s, data, elem) BOOST_PP_CAT(dpt_, elem)

Typedefs

- typedef [dpt_1_001](#) [knx::dpt_switch](#)
data point type 1.001 is better known as switch
- typedef [dpt_5_001](#) [knx::dpt_scaling](#)
5.001 is better known as scaling value
- typedef [dpt_10_001](#) [knx::dpt_timeofday](#)
10.001 is known as time of day value including hours. minutes, seconds and day of the week
- typedef `boost::mpl::vector<>` [knx::dpt_types](#)

Enumerations

- enum [dpt_index](#)

Functions

- `std::ostream & knx::operator<< (std::ostream &os, const dpt_10::time &time)`

7.4.1 Detailed Description

All available dpts are defined in this file.

7.4.2 Macro Definition Documentation

7.4.2.1 #define SUPPORTED_DPTS

Value:

```
(1_001) \
(2_001) \
(3_007) \
(4_001) \
(5_001) \
(6_001) \
(7_001) \
(8_001) \
(9_001) \
(10_001) \
(11_001) \
(12_001) \
(13_001) \
(14_000) \
(15_000) \
(16_000) \
(17_001) \
(18_001) \
(19_001) \
(20_011) \
(21_001) \
(26_001) \
(219_001) \
(232_600)
```

7.5 group.hpp File Reference

```
#include <inttypes.h>
#include <string>
#include <sstream>
```

Classes

- class [knx::group](#)

This class represents a KNX group address which might be either 3 component "a/b/c" or 2 component "a/b".

Namespaces

- [knx](#)

separated namespace to embed all libknx related classes

7.5.1 Detailed Description

Class file for group address handling.

7.6 handle.hpp File Reference

```
#include <boost/thread.hpp>
#include "debug.hpp"
#include "connection.hpp"
```

Classes

- class [knx::handle](#)
Thread management for connection handling.

Namespaces

- [knx](#)
separated namespace to embed all libknx related classes

7.6.1 Detailed Description

This file contains the helper class for handling knx access.

7.7 knx.hpp File Reference

```
#include "config.hpp"
#include "connection.hpp"
#include "handle.hpp"
#include "dpt.hpp"
#include "data_point.hpp"
```

7.7.1 Detailed Description

General include header for libknx.

Index

knx, [11](#)