libknx

Generated by Doxygen 1.8.1.2

Wed Aug 7 2013 01:37:55

Contents

Chapter 1

KNX interface library

Author

Norbert Schmitz, knx@nagilo.de

Date

July 2013

Version

1.0.0

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 .

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>("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 compiltion 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 1.0.0 has the following limitations:

- Only 3-part group addresses are available
- Only UDP communication is available
- Only data point types 1.001(rw), 5.001(r) amd 10.001(w) are available

Ideas

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

Yours Norbert Schmitz

Chapter 2

Namespace Index

2.1 Namespace List

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

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 handle

    thread management for connection handling
- class dpt
- class dpt_1

    major class for all 1.XXX data point types
- class dpt_1_001

    data point type 1.001 simple boolean value
- class dpt_5

    major class for all 5.XXX data point types
- class dpt_5_001

    data point type 5.001 scaling value
- class dpt_10

    major class for all 10.XXX data point types
- class dpt_10_001

    class holding a time value
- class data_point

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

Functions

- template<class dpt >
  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 descructor

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 (std::string 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 (std::string group, typename data_type::major_type::set_type &data)

    central function to 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 (std::string *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

group	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 > void knx::connection::set (std::string *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.

6.2.2.3 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.

6.2.2.4 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::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

6.4 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

6.5 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)

6.5.1 Detailed Description

major class for all 1.XXX data point types

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

- dpt.hpp

6.6 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)

6.6.1 Detailed Description

major class for all 10.XXX data point types

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

- dpt.hpp

6.7 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 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

6.7.1 Detailed Description

class holding a time value

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

- dpt.hpp

6.8 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 major_type::set_type **ON** = true
- static const major_type::set_type **OFF** = false

6.8.1 Detailed Description

data point type 1.001 simple boolean value

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

- dpt.hpp

6.9 knx::dpt_5 Class Reference

major class for all 5.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)

6.9.1 Detailed Description

major class for all 5.XXX data point types

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

- dpt.hpp

6.10 knx::dpt_5_001 Class Reference

data point type 5.001 scaling value

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

Public Types

- typedef dpt_5 **major_type**

Static Public Member Functions

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

6.10.1 Detailed Description

data point type 5.001 scaling value

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

- dpt.hpp

6.11 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

6.11.1 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

6.12 knx::dpt_10::time Class Reference

internal class storing a time value

`#include <dpt.hpp>`

Public Member Functions

- time ()

  creates a current time value

Public Attributes

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

6.12.1 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 knx.hpp File Reference

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

7.1.1 Detailed Description

General include header for libknx.

Index