SZS_soundzone_protocol Doxy-Documentation

Generated by Doxygen 1.9.2

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 buffer_t Class Reference

Public Member Functions

- **buffer_t** (uint8_t ∗buffer, uint16_t size)
  *Initializes the **buffer_t** (p. 7) object.*
- int **append** (uint8_t byte)
  Appends one byte at the write head.
- int **append** (long long int value)
  Appends one 'long long int' at the write head.
- int **append** (const uint8_t ∗buffer, uint16_t bytes_to_write)
  Appends an array at the write head.
- int **read_one** (uint8_t ∗value)
- int **read_one** (long long int ∗value)
- int **get_buffer_rest** (uint8_t ∗∗buffer, uint16_t ∗size)
  *Used to get the rest of the unread **buffer_t** (p. 7).*
- int **reset** ()
  Resets read and write head.
- int **set_write_head** (uint16_t head)
  Sets the write head index and resets the read head index.
- uint16_t **get_write_head** ()
  Getter of write head index.
- int **print_buffer** ()
  Prints the buffer using cout.

4.1.1 Constructor & Destructor Documentation

4.1.1.1 buffer_t()

```
buffer_t::buffer_t (
            uint8_t * buffer,
            uint16_t size )
```

Initializes the **buffer_t** (p. 7) object.

Parameters

buffer[in]	Pointer to the array that the **buffer_t** (p. 7) should use.
size[in]	Size of #buffer.

4.1.2 Member Function Documentation

4.1.2.1 append() [1/3]

```
int buffer_t::append (
            const uint8_t * buffer,
            uint16_t bytes_to_write )
```

Appends an array at the write head.

If #buffer is bigger than there is space in **buffer_t** (p. 7) is will not write any, and just return error.

Parameters

buffer[in]	The array to append.
bytes_to_write[in]	The number of bytes to write from #buffer.

Returns

int

Return values

0	If successful
-1	If Buffer is full.

4.1.2.2 append() [2/3]

```
int buffer_t::append (
            long long int value )
```

Appends one 'long long int' at the write head.

Parameters

value[in]	The int to append.

Returns

int

Return values

0	If successful
-1	If the buffer is full.

4.1.2.3 append() [3/3]

```
int buffer_t::append (
            uint8_t byte )
```

Appends one byte at the write head.

Parameters

byte[in]	The byte to append.

Returns

int

Return values

0	If successful
-1	If the buffer is full.

4.1.2.4 get_buffer_rest()

```
int buffer_t::get_buffer_rest (
            uint8_t ** buffer,
            uint16_t * size )
```

Used to get the rest of the unread **buffer_t** (p. 7).

Parameters

buffer[out]	Pointer to the read head.
size[out]	The size of the rest of the **buffer_t** (p. 7).

Returns

int

Return values

0	If successful.
-2	If buffer i empty.

4.1.2.5 get_write_head()

```
uint16_t buffer_t::get_write_head ( )
```

Getter of write head index.

Returns

Write head index.

4.1.2.6 print_buffer()

```
int buffer_t::print_buffer ( )
```

Prints the buffer using cout.

Mainly used for debugging, as it has no effect on read or write head or the array it self.

Returns

0.

4.1.2.7 read_one() [1/2]

```
int buffer_t::read_one (
            long long int * value )
```

Reads one 'long long int' from **buffer_t** (p. 7).

Parameters

value[uot]	Pointer to where to put the read int.

Returns

int

Return values

0	If successful.
-3	If **buffer_t** (p. 7) is empty.

4.1.2.8 read_one() [2/2]

```
int buffer_t::read_one (
            uint8_t * value )
```

Reads one byte from **buffer_t** (p. 7).

Parameters

byte[uot]	Pointer to where to put the read byte.

Returns

int

Return values

0	If successful.
-3	If **buffer_t** (p. 7) is empty.

4.1.2.9 reset()

```
int buffer_t::reset ( )
```

Resets read and write head.

Returns

0;

4.1.2.10 set_write_head()

```
int buffer_t::set_write_head (
            uint16_t head )
```

Sets the write head index and resets the read head index.

Parameters

head[in]	Write head index.

Returns

0.

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

- C:/Users/ncpd/Documents/GitHub/szs_soundzone_protocol/include/szp_custom_types.h

4.2 debug_write_file Class Reference

Public Member Functions

- **debug_write_file** ()
- ∼**debug_write_file** ()
- void **write** (const uint8_t ∗buff, int size)

  Opens, appends and closes to /tmp/rcv_file.wav.

4.2.1 Constructor & Destructor Documentation

4.2.1.1 debug_write_file()

```
debug_write_file::debug_write_file ( )
```

Resets the /tmp/rcv_file.wav

4.2.1.2 ∼debug_write_file()

```
debug_write_file::∼debug_write_file ( )
```

Closes debug_fd.

4.2.2 Member Function Documentation

4.2.2.1 write()

```
void debug_write_file::write (
            const uint8_t * buff,
            int size )
```

Opens, appends and closes to /tmp/rcv_file.wav.

Parameters

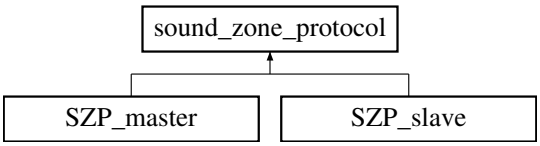
buff[in]	Buffer to append.
size[in]	size of #buff

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

- C:/Users/ncpd/Documents/GitHub/szs_soundzone_protocol/include/debug_methods.h

4.3 sound_zone_protocol Class Reference

Inheritance diagram for sound_zone_protocol:



Public Member Functions

- **sound_zone_protocol** ()

  *Default constructor for **sound_zone_protocol** (p. 13).*
- **sound_zone_protocol** (uint8_t ∗comm_buffer, uint16_t buffer_size)

  *Constructor for **sound_zone_protocol** (p. 13).*
- int **set_fifo** (int ∗fifo_fd)

  Sets fifo file descriptor.
- int **set_values** (uint8_t value)

  Sets the package values for the command defined by #cid.
- int **set_values** (long long int value)

  Sets the package values for the command defined by #cid.
- int **set_values** (uint8_t ∗values, uint16_t size)

  Sets the package values for the command defined by #cid.
- **buffer_t** ∗ **encode** (**buffer_t** ∗encoded_msg)

  *Serializes the **sound_zone_protocol** (p. 13) object.*
- int **decode** (**buffer_t** ∗msg_to_decode)

Protected Member Functions

- int **get_values** (long long int ∗value)

  Get the package values for the command defined by #cid.
- uint16_t **encode_and_get_size** ()

  Serializes the SZP object and returns the size.

Protected Attributes

- **buffer_t** * **p_buffer**
- supported_cid_t **cid**
- **xF1_check_connection** * **check_connection**
- **x01_send_sound_packet** * **send_sound_packet**

4.3.1 Constructor & Destructor Documentation

4.3.1.1 sound_zone_protocol()

```
sound_zone_protocol::sound_zone_protocol (
            uint8_t * comm_buffer,
            uint16_t buffer_size )
```

Constructor for **sound_zone_protocol** (p. 13).

This constructor gives to possibility to use an external communication buffer.

Parameters

comm_buffer[in]	Pointer to the external communication buffer.
buffer_size[in]	Size of #comm_buffer.

4.3.2 Member Function Documentation

4.3.2.1 decode()

```
int sound_zone_protocol::decode (
             buffer_t * msg_to_decode )
```

Decodes a serialized SZP object.

Parameters

msg_to_decode[in]	Pointer to the serialized object.

Returns

int.

Return values

0	If successful.
-1	If #cid not supported.

4.3.2.2 encode()

```
buffer_t * sound_zone_protocol::encode (
            buffer_t * encoded_msg )
```

Serializes the **sound_zone_protocol** (p. 13) object.

Parameters

encoded_msg[out]	Pointer to the **buffer_t** (p. 7) where the serialized SZP object is saved.

Returns

#encoded_msg.

4.3.2.3 encode_and_get_size()

```
uint16_t sound_zone_protocol::encode_and_get_size ( )  [protected]
```

Serializes the SZP object and returns the size.

Returns

uint16_t - size of serialized object.

4.3.2.4 get_values()

```
int sound_zone_protocol::get_values (
            long long int * value )  [protected]
```

Get the package values for the command defined by #cid.

Depending on the #cid value is filled.

Parameters

value[in]	the LL int value to set.

Returns

int.

Return values

0	If successful.
-1	If #cid is not set.
-2	If the #cid does not support this type of package value.
-3	If the get_values sub method returns error.

4.3.2.5 set_fifo()

```
int sound_zone_protocol::set_fifo (
            int * fifo_fd )
```

Sets fifo file descriptor.

Parameters

fifo_fd[in]	Fifo file descriptor.

Returns

0

4.3.2.6 set_values() [1/3]

```
int sound_zone_protocol::set_values (
            long long int value )
```

Sets the package values for the command defined by #cid.

Depending on the #cid values are filled.

Parameters

value[in]	the byte value to set.

Returns

int.

Return values

0	If successful.
-1	If #cid is not set.
-2	If the #cid does not support this type of package value.

4.3.2.7 set_values() [2/3]

```
int sound_zone_protocol::set_values (
            uint8_t * values,
            uint16_t size )
```

Sets the package values for the command defined by #cid.

Depending on the #cid values are filled.

Parameters

values[in]	Pointer to the array of values to set.
size[in]	The size of #values, in bytes.

Returns

int.

Return values

0	If successful.
-1	If #cid is not set.
-2	If the #cid does not support this type of package value.

4.3.2.8 set_values() [3/3]

```
int sound_zone_protocol::set_values (
            uint8_t value )
```

Sets the package values for the command defined by #cid.

Depending on the #cid values are filled.

Parameters

value[in]	the byte value to set.

Returns

int.

Return values

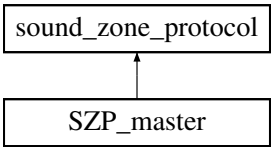
0	If successful.
-1	If #cid is not set.
-2	If the #cid does not support this type of package value.

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

- C:/Users/ncpd/Documents/GitHub/szs_soundzone_protocol/include/sound_zone_protocol.h

4.4 SZP_master Class Reference

Inheritance diagram for SZP_master:



Public Member Functions

- **SZP_master** ()
- **SZP_master** (char ∗host, bool is_ip)

  *Constructor for **SZP_master** (p. 18), host is used to specify the **SZP_slave** (p. 21).*
- int **check_connection** ()

  Does what it says.
- int **send_sound_packet** (uint8_t ∗buffer, uint16_t packet_size, long long int play_time)

  Sends a SZP sound package to the host.

Additional Inherited Members

4.4.1 Constructor & Destructor Documentation

4.4.1.1 SZP_master() [1/2]

```
SZP_master::SZP_master ( )
```

Default constructor for **SZP_master** (p. 18)

4.4.1.2 SZP_master() [2/2]

```
SZP_master::SZP_master (
            char * host,
            bool is_ip )
```

Constructor for **SZP_master** (p. 18), host is used to specify the **SZP_slave** (p. 21).

Note

Support for hostname not implemented yet.

Parameters

host[in]	Ip or hostname of host.
is_ip[in]	Defines if #host is an IP

4.4.2 Member Function Documentation

4.4.2.1 check_connection()

```
int SZP_master::check_connection ( )
```

Does what it says.

Checks connection by sending a SZP **xF1_check_connection** (p. 26) to the host. If host returns an ack connection is running.

Returns

int

Return values

0	if good
-1	if no ack.

4.4.2.2 send_sound_packet()

```
int SZP_master::send_sound_packet (
            uint8_t * buffer,
            uint16_t packet_size,
            long long int play_time )
```

Sends a SZP sound package to the host.

Parameters

buffer[in]	Pointer to the sound block to send.
packet_size[in]	Size of the sound block.
play_time[in]	The time the sound block sound be played.

Returns

int

Return values

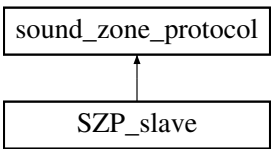
0	If successful
-1	If send failed.

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

- C:/Users/ncpd/Documents/GitHub/szs_soundzone_protocol/include/szp_master.h

4.5 SZP_slave Class Reference

Inheritance diagram for SZP_slave:



Public Member Functions

- **SZP_slave** (char ∗fifo_name)
  *Constructor of **SZP_slave*** (p. 21).
- ∼**SZP_slave** ()
  Does nothing.
- int **open_fifo** ()
  Opens the fifo file in write only mode. Hangs until read side is opened.
- int **recieve** ()
  Receives a package from DataTransport and the reacts on it.
- int **get_time** (long long int ∗time)
  Getter of #time_to_play.

Additional Inherited Members

4.5.1 Constructor & Destructor Documentation

4.5.1.1 SZP_slave()

```
SZP_slave::SZP_slave (
            char * fifo_name )  [explicit]
```

Constructor of **SZP_slave** (p. 21).

Parameters

in	*fifo_name*	Name and path of the fifo buffer to use under x01_send_sound_package.

4.5.2 Member Function Documentation

4.5.2.1 get_time()

```
int SZP_slave::get_time (
            long long int * time )
```

Getter of #time_to_play.

Parameters

time	Pointer where to store the value of #time_to_play.

Returns

int

Return values

0	If successful.
-1	If time invalid.

4.5.2.2 open_fifo()

```
int SZP_slave::open_fifo ( )
```

Opens the fifo file in write only mode. Hangs until read side is opened.

Returns

0.

4.5.2.3 recieve()

```
int SZP_slave::recieve ( )
```

Receives a package from DataTransport and the reacts on it.

Depending on the received package it also responds.

Returns

0

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

- C:/Users/ncpd/Documents/GitHub/szs_soundzone_protocol/include/szp_slave.h

4.6 x01_send_sound_packet Class Reference

Public Member Functions

- **x01_send_sound_packet** ()
- int **set_fifo** (const int *fifo_fd)
  Sets the fifo fd.
- int **set_values** (long long int value)
  Sets the #time value.
- int **set_values** (uint8_t *values, uint16_t size)
  Sets the package values for the command.
- int **get_values** (long long int *value)
- **buffer_t** * **encode** (**buffer_t** *encoded_msg)
  Serializes the x01 object.
- void **decode** (**buffer_t** *msg_to_decode)
  Decodes a x01 serialized object.
- int **reset** ()
  Resets it #p_payload.

4.6.1 Constructor & Destructor Documentation

4.6.1.1 x01_send_sound_packet()

```
x01_send_sound_packet::x01_send_sound_packet ( )
```

Default constructor.

4.6.2 Member Function Documentation

4.6.2.1 decode()

```
void x01_send_sound_packet::decode (
            buffer_t * msg_to_decode )
```

Decodes a x01 serialized object.

Parameters

msg_to_decode[in]	Pointer to the buffer containing the serialized x01 obj.

4.6.2.2 encode()

```
buffer_t * x01_send_sound_packet::encode (
            buffer_t * encoded_msg )
```

Serializes the x01 object.

Parameters

encoded_msg[out]	The **buffer_t** (p. 7) object where the serialized x01 object is saved.

Returns

#encoded_msg.

4.6.2.3 get_values()

```
int x01_send_sound_packet::get_values (
            long long int * value )
```

Getter of time value.

Parameters

value	Pointer where to store time value.

Returns

int

Return values

0	If successful.
-1	If time is not set.

4.6.2.4 reset()

```
int x01_send_sound_packet::reset ( )
```

Resets it #p_payload.

Returns

0

4.6.2.5 set_fifo()

```
int x01_send_sound_packet::set_fifo (
            const int * fifo_fd )
```

Sets the fifo fd.

Parameters

fifo_fd[in]	Fifo file descriptor.

Returns

0

4.6.2.6 set_values() [1/2]

```
int x01_send_sound_packet::set_values (
            long long int value )
```

Sets the #time value.

value/time cannot be negative.

Parameters

value[in]	Time value in micro-seconds.

Returns

int

Return values

0	If successful
-1	If value has wrong format.

4.6.2.7 set_values() [2/2]

```
int x01_send_sound_packet::set_values (
            uint8_t * values,
            uint16_t size )
```

Sets the package values for the command.

Parameters

values[in]	Pointer to the array of values to set.
size[in]	The size of #values, in bytes.

Returns

0

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

- C:/Users/ncpd/Documents/GitHub/szs_soundzone_protocol/include/x01_send_sound_packet.h

4.7 xF1_check_connection Class Reference

Public Member Functions

- **xF1_check_connection** ()
  Default constructor.
- int **set_values** (uint8_t value)
  Sets the #acknolegment value.
- int **reset** ()
  Resets #acknolegment.
- **buffer_t** * **encode** (**buffer_t** *encoded_msg)
  Serializes the xF1 object.
- int **decode** (**buffer_t** *buffer)
  Decodes a serialized xF1 object.

4.7.1 Member Function Documentation

4.7.1.1 decode()

```
int xF1_check_connection::decode (
            buffer_t * buffer )
```

Decodes a serialized xF1 object.

Parameters

msg_to_decode[in]	Pointer to the buffer containing the serialized xF1 obj.

4.7.1.2 encode()

```
buffer_t * xF1_check_connection::encode (
            buffer_t * encoded_msg )
```

Serializes the xF1 object.

Parameters

encoded_msg[out]	The **buffer_t** (p. 7) where the serialized xF1 object is saved.

Returns

#encoded_msg.

4.7.1.3 reset()

```
int xF1_check_connection::reset ( )
```

Resets #acknolegment.

Returns

0

4.7.1.4 set_values()

```
int xF1_check_connection::set_values (
            uint8_t value )
```

Sets the #acknolegment value.

Parameters

value[in]	The ack byte.

Returns

0

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

- C:/Users/ncpd/Documents/GitHub/szs_soundzone_protocol/include/xF1_check_connection.h

Chapter 5

File Documentation

5.1 debug_methods.h

```
1
      /*****************************************************************************************************************************************
2   * @name Debug methods
3   * @file debug_methods.h
4   * @author Niels Dyrberg
5   * @date 17-11-2021
6   *
7   * Purpose:
8   *       Should holds the classes and method used for debugging, or patch/temporary solutions.
9
      *****************************************************************************************************************************************/
10
11 #ifndef SOUNDZONE_CLIENT_DEBUG_METHODS_H
12 #define SOUNDZONE_CLIENT_DEBUG_METHODS_H
13
14
      /*****************************************************************************************************************************************
15   * Includes
16
      *****************************************************************************************************************************************/
17
18 #include <cstdio>
19 #include <cstdlib>
20 #include <cstdint>
21 #include <fstream>
22
23
      /*****************************************************************************************************************************************
24   * debug_write_file class declaration
25
      *****************************************************************************************************************************************/
26
27 class debug_write_file {
28 private:
29     std::ofstream debug_fd;
30
31 public:
35     debug_write_file();
36
40     ~debug_write_file();
41
47     void write(const uint8_t *buff, int size);
48 };
49
50 #endif //SOUNDZONE_CLIENT_DEBUG_METHODS_H
```

5.2 sound_zone_protocol.h

```
1
      /*****************************************************************************************************************************************
2   * @name Sound-Zone Protocol
3   * @file sound_zone_protocol.h
4   * @author Niels Dyrberg
5   * @date 27-10-2021
```

```
6  *
7  * Purpose:
8  *      Encode and decode SZP packages.
9  *
10  * @note Is written as a base-class for SZP_master and SZP_slave.
11
       ********************************************************************************************************************************/
12
13 #ifndef C_SOUNDZONE_CLIENT_SOUND_ZONE_PROTOCOL_H
14 #define C_SOUNDZONE_CLIENT_SOUND_ZONE_PROTOCOL_H
15
16
       /********************************************************************************************************************************
17  * Includes
18
       ********************************************************************************************************************************/
19
20 #include <iostream>
21 #include <cstdint>
22
23 #include "dataTransport.h"
24 #include "xF1_check_connection.h"
25 #include "x01_send_sound_packet.h"
26 #include "szp_custom_types.h"
27
28
       /********************************************************************************************************************************
29  * Class declaration
30
       ********************************************************************************************************************************/
31
32 class sound_zone_protocol {
33 public:
37     sound_zone_protocol();
38
45     sound_zone_protocol(uint8_t *comm_buffer, uint16_t buffer_size);
46
52     int set_fifo(int *fifo_fd);
53
63     int set_values(uint8_t value);
64
74     int set_values(long long int value);
75
86     int set_values(uint8_t *values, uint16_t size);
87
93     buffer_t *encode(buffer_t *encoded_msg);
94
102     int decode(buffer_t *msg_to_decode);
103
104 protected:
105     buffer_t *p_buffer;
106     supported_cid_t cid;
107     xF1_check_connection *check_connection;
108     x01_send_sound_packet *send_sound_packet;
109
120     int get_values(long long int *value);
121
126     uint16_t encode_and_get_size();
127
128 private:
129     static supported_cid_t initial_decode(uint8_t cid);
130 };
131
132 #endif //C_SOUNDZONE_CLIENT_SOUND_ZONE_PROTOCOL_H
```

5.3 szp_custom_types.h

```
1
       /********************************************************************************************************************************
2  * @name SZP custom types
3  * @file szp_custom_types.h
4  * @author Niels Dyrberg
5  * @date 27-10-2021
6  *
7  * Purpose:
8  *      Ease up data movement between classes in the SZP package.
9
       ********************************************************************************************************************************/
10
11 #ifndef C_SOUNDZONE_CLIENT_SZP_CUSTOM_TYPES_H
12 #define C_SOUNDZONE_CLIENT_SZP_CUSTOM_TYPES_H
13
14
       /********************************************************************************************************************************
```

```
15  * Includes
16
    ********************************************************************************************************************************/
17
18 #include <cstdint>
19
20
    /********************************************************************************************************************************
21  * Typedef of supported_cid_t
22
    ********************************************************************************************************************************/
23
24 enum supported_cid_t : uint8_t {
25     cid_send_sound_packet = 1,
26     cid_enroll = 161,
27     cid_set_sound_format = 179,
28     cid_check_connection = 241,
29     cid_notSet = 255
30 };
31
32
    /********************************************************************************************************************************
33  * Class declaration
34
    ********************************************************************************************************************************/
35
36 class buffer_t {
37 public:
43     buffer_t(uint8_t *buffer, uint16_t size);
44
52     int append(uint8_t byte);
53
61     int append(long long int value);
62
72     int append(const uint8_t *buffer, uint16_t bytes_to_write);
73
81     int read_one(uint8_t *value);
82
90     int read_one(long long int* value);
91
100     int get_buffer_rest(uint8_t **buffer, uint16_t *size);
101
106     int reset();
107
113     int set_write_head(uint16_t head);
114
119     uint16_t get_write_head();
120
126     int print_buffer();
127
128 private:
129     uint8_t *p_buffer;
130     uint16_t buffer_size;
131     uint16_t write_head;
132     uint16_t read_head;
133 protected:
134 };
135
136 #endif //C_SOUNDZONE_CLIENT_SZP_CUSTOM_TYPES_H
```

5.4 szp_master.h

```
1
    /********************************************************************************************************************************
2   * @name SZP_master
3   * @file szp_master.h
4   * @author Niels Dyrberg
5   * @date 08-11-2021
6   *
7   * Purpose:
8   *      Encode/decodes sent/received SZP packages.
9
    ********************************************************************************************************************************/
10
11 #ifndef SZS_SOUNDZONE_PROTOCOL_SZP_MASTER_H
12 #define SZS_SOUNDZONE_PROTOCOL_SZP_MASTER_H
13
14
    /********************************************************************************************************************************
15  * Includes
16
    ********************************************************************************************************************************/
17
```

```
18 #include <fstream>
19 #include "sound_zone_protocol.h"
20 #include "udp_client.h"
21
22
       /*****************************************************************************************************************
23  * Defines
24
       *****************************************************************************************************************/
25
26 #define COMM_BUFFER_SIZE 4096
27
28
       /*****************************************************************************************************************
29  * Class declaration
30
       *****************************************************************************************************************/
31
32 class SZP_master : public sound_zone_protocol {
33 public:
37     SZP_master();
38
45     SZP_master(char *host, bool is_ip);
46
55     int check_connection();
56
66     int send_sound_packet(uint8_t *buffer, uint16_t packet_size, long long int play_time);
67
68 protected:
69
70
71 private:
72     UDP_client dt;
73     uint8_t comm_buffer[COMM_BUFFER_SIZE] = {};
74
75
76 };
77
78 #endif //SZS_SOUNDZONE_PROTOCOL_SZP_MASTER_H
```

5.5 szp_slave.h

```
1
       /*****************************************************************************************************************
2  * @name SZP_slave
3  * @file szp_slave.h
4  * @author Niels Dyrberg
5  * @date 08-11-2021
6  *
7  * Purpose:
8  *      Encode/decodes sent/received SZP packages.
9
       *****************************************************************************************************************/
10
11 #ifndef SZS_SOUNDZONE_PROTOCOL_SZP_SLAVE_H
12 #define SZS_SOUNDZONE_PROTOCOL_SZP_SLAVE_H
13
14
       /*****************************************************************************************************************
15  * Includes
16
       *****************************************************************************************************************/
17
18 #include "sound_zone_protocol.h"
19 #include "udp_server.h"
20
21
       /*****************************************************************************************************************
22  * Defines
23
       *****************************************************************************************************************/
24
25 #define COMM_BUFFER_SIZE 4096
26
27
       /*****************************************************************************************************************
28  * Class declaration
29
       *****************************************************************************************************************/
30
31 class SZP_slave : public sound_zone_protocol {
32 public:
37     explicit SZP_slave(char *fifo_name);
```

```
38
42      ~SZP_slave();
43
48      int open_fifo();
49
55      int recieve();
56
64      int get_time(long long int *time);
65
66  protected:
67
68  private:
69      UDP_server dt;
70      char *fifo_name;
71      uint8_t comm_buffer[COMM_BUFFER_SIZE] = {};
72      long long int time_to_play;
73
80      int encode_and_send();
81
88      int react_on_incoming();
89  };
90
91
92  #endif //SZS_SOUNDZONE_PROTOCOL_SZP_SLAVE_H
```

5.6 x01_send_sound_packet.h

```
1
        /*****************************************************************************************************************************
2   * @name x01 Send sound-packet
3   * @file x01_send_sound_packet.h
4   * @author Niels Dyrberg
5   * @date 27-10-2021
6   *
7   * Purpose:
8   *      Encode and decode SZP x01-send-sound-packages.
9
        *****************************************************************************************************************************/
10
11  #ifndef C_SOUNDZONE_CLIENT_X01_SEND_SOUND_PACKET_H
12  #define C_SOUNDZONE_CLIENT_X01_SEND_SOUND_PACKET_H
13
14
        /*****************************************************************************************************************************
15   * Includes
16
        *****************************************************************************************************************************/
17
18  #include <iostream>
19  #include <cstdint>
20  #include "szp_custom_types.h"
21
22
        /*****************************************************************************************************************************
23   * Defines
24
        *****************************************************************************************************************************/
25
26  #define TIME_NOT_SET (long long int)-1
27
28
        /*****************************************************************************************************************************
29   * Class declaration
30
        *****************************************************************************************************************************/
31
32  class x01_send_sound_packet {
33  private:
34      uint8_t *p_payload;
35      uint16_t payload_size;
36      int fifo_fd;
37      static long long int time;
38
39  public:
43      x01_send_sound_packet();
44
50      int set_fifo(const int *fifo_fd);
51
60      int set_values(long long int value);
61
68      int set_values(uint8_t *values, uint16_t size);
69
77      int get_values(long long int *value);
```

```
78
84      buffer_t *encode(buffer_t *encoded_msg);
85
90      void decode(buffer_t *msg_to_decode);
91
96      int reset();
97  };
98
99  #endif //C_SOUNDZONE_CLIENT_X01_SEND_SOUND_PACKET_H
```

5.7 xF1_check_connection.h

```
1
        /*****************************************************************************************************************************
2   * @name xF1 Check connection
3   * @file xf1_check_connection.h
4   * @author Niels Dyrberg
5   * @date 27-10-2021
6   *
7   * Purpose:
8   *       Encode and decode SZP xf1-check-connection.
9
        *****************************************************************************************************************************/
10
11  #ifndef C_SOUNDZONE_CLIENT_XF1_CHECK_CONNECTION_H
12  #define C_SOUNDZONE_CLIENT_XF1_CHECK_CONNECTION_H
13
14
        /*****************************************************************************************************************************
15   * Includes
16
        *****************************************************************************************************************************/
17
18  #include <cstdint>
19  #include "szp_custom_types.h"
20
21
        /*****************************************************************************************************************************
22   * Class declaration
23
        *****************************************************************************************************************************/
24
25  class xF1_check_connection {
26  public:
30      xF1_check_connection();
31
37      int set_values(uint8_t value);
38
43      int reset();
44
50      buffer_t *encode(buffer_t *encoded_msg);
51
56      int decode(buffer_t *buffer);
57
58  protected:
59  private:
60      uint8_t acknowledgment;
61
62  };
63
64  #endif //C_SOUNDZONE_CLIENT_XF1_CHECK_CONNECTION_H
```

Index