

UDP

Generated by Doxygen 1.16.1

1 udp	1
2 Topic Index	3
2.1 Topics	3
3 Directory Hierarchy	5
3.1 Directories	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Topic Documentation	11
6.1 work for udp	11
6.1.1 Detailed Description	12
6.1.2 Typedef Documentation	12
6.1.2.1 udp_pack_t	12
6.1.3 Function Documentation	13
6.1.3.1 add_byte_udp_pack()	13
6.1.3.2 add_date_udp_pack()	14
6.1.3.3 calculate_checksum_udp_pack()	15
6.1.3.4 checksum_compute()	15
6.1.3.5 destroy_udp_pack()	15
6.1.3.6 get_pack_udp_pack()	16
6.1.3.7 get_size_data_udp_pack()	16
6.1.3.8 inet_port()	16
6.1.3.9 init_udp_pack()	17
6.1.3.10 send_udp_pack()	17
6.1.3.11 set_data_udp_pack()	18
6.1.3.12 set_file_data_udp_pack()	18
6.1.3.13 set_input_data_udp_pack()	19
6.1.3.14 set_interface_udp_pack()	19
6.1.3.15 set_ip_destination_udp_pack()	20
6.1.3.16 set_ip_source_udp_pack()	20
6.1.3.17 set_port_destination_udp_pack()	21
6.1.3.18 set_port_source_udp_pack()	21
6.1.3.19 set_size_udp_pack()	22
6.1.3.20 sum_compute()	22
7 Directory Documentation	23
7.1 udp_lib Directory Reference	23
8 Class Documentation	25

8.1 pseudo_header Struct Reference	25
8.1.1 Detailed Description	25
8.1.2 Member Data Documentation	25
8.1.2.1 dest_address	25
8.1.2.2 placeholder	25
8.1.2.3 protocol	26
8.1.2.4 source_address	26
8.1.2.5 udp_length	26
8.2 udp_head Struct Reference	26
8.2.1 Detailed Description	26
8.2.2 Member Data Documentation	27
8.2.2.1 m_checksum	27
8.2.2.2 m_length	27
8.2.2.3 m_port_destination	27
8.2.2.4 m_port_source	27
8.3 udp_pack Struct Reference	27
8.3.1 Detailed Description	28
8.3.2 Member Data Documentation	28
8.3.2.1 m_data	28
8.3.2.2 m_ethhdr	28
8.3.2.3 m_head	28
8.3.2.4 m_interface	28
8.3.2.5 m_iphdr	28
9 File Documentation	29
9.1 main.c File Reference	29
9.1.1 Function Documentation	29
9.1.1.1 main()	29
9.2 README.md File Reference	29
9.3 udp_lib/udp.c File Reference	29
9.3.1 Detailed Description	31
9.3.2 Macro Definition Documentation	31
9.3.2.1 HEAD_ETH	31
9.3.2.2 HEAD_IP	31
9.3.2.3 HEAD_PSEUDO	31
9.3.2.4 HEAD_UDP	32
9.3.2.5 HEAD_UDP_IP	32
9.3.2.6 MAX_SIZE_DATA	32
9.3.2.7 MIN	32
9.3.2.8 NULL_CHECKSUM	32
9.3.2.9 PACKED	32
9.4 udp_lib/udp.h File Reference	32

9.4.1 Detailed Description	33
9.5 udp.h	33
Index	35

Chapter 1

udp

Chapter 2

Topic Index

2.1 Topics

Here is a list of all topics with brief descriptions:

work for udp [11](#)

Chapter 3

Directory Hierarchy

3.1 Directories

udp_lib	23
udp.c	29
udp.h	32

Chapter 4

Class Index

4.1 Class List

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

pseudo_header		
	Struct pseudo_header for calculate checksum for UDP package	25
udp_head		
	Struct is header UDP pack	26
udp_pack		
	Struct is UDP package	27

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

main.c	29
udp_lib/udp.c	
Code file for work udp package	29
udp_lib/udp.h	
Header file for work udp package	32

Chapter 6

Topic Documentation

6.1 work for udp

Group function for work udp package.

Classes

- struct [udp_head](#)
Struct is header UDP pack.
- struct [udp_pack](#)
Struct is UDP package.
- struct [pseudo_header](#)
Struct [pseudo_header](#) for calculate checksum for UDP package.

Typedefs

- typedef struct [udp_pack](#) * [udp_pack_t](#)
UDP packet descriptor.

Functions

- static uint16_t [inet_port](#) (const char *const port)
Function support for parsing port.
- static void [set_size_udp_pack](#) ([udp_pack_t](#) pack, const uint16_t size)
Function raw write size UDP package.
- static uint32_t [sum_compute](#) (void *ptr, uint16_t nbytes)
Function calculate sum in big endian.
- static uint16_t [checksum_compute](#) (uint32_t sum)
Function calculating from sum big endian to checksum big endian.
- static void [calculate_checksum_udp_pack](#) ([udp_pack_t](#) pack)
Function calculate checksum for ip header and UDP package.
- static void * [get_pack_udp_pack](#) ([udp_pack_t](#) pack)
Function raw get pointer on UDP package.
- [udp_pack_t](#) [init_udp_pack](#) (void)

- Function for create object UDP package.*

 - ssize_t [set_port_source_udp_pack](#) ([udp_pack_t](#) pack, const char *const port)

Function for setting source port in UDP package.
- ssize_t [set_port_destination_udp_pack](#) ([udp_pack_t](#) pack, const char *const port)

Function for setting destination port in UDP package.
- ssize_t [set_ip_source_udp_pack](#) ([udp_pack_t](#) pack, const char *const ip)

Function for setting source ip in UDP package.
- ssize_t [set_ip_destination_udp_pack](#) ([udp_pack_t](#) pack, const char *const ip)

Function for setting destination ip in UDP package.
- uint16_t [get_size_data_udp_pack](#) ([udp_pack_t](#) pack)

Function give size data in UDP package.
- ssize_t [add_data_udp_pack](#) ([udp_pack_t](#) pack, void *data, uint16_t size)

Function addition data in UDP package.
- ssize_t [set_data_udp_pack](#) ([udp_pack_t](#) pack, void *data, const uint16_t size)

Function overriding data in UDP package.
- ssize_t [add_byte_udp_pack](#) ([udp_pack_t](#) pack, const uint8_t byte)

Function addition byte in UDP package.
- ssize_t [set_input_data_udp_pack](#) ([udp_pack_t](#) pack)

Function read data from stdin in UDP package.
- ssize_t [set_file_data_udp_pack](#) ([udp_pack_t](#) pack, const char *const file_name)

Function read data from file in UDP package.
- ssize_t [set_interface_udp_pack](#) ([udp_pack_t](#) pack, const char *const interface)

Function for pick interface to send UDP package.
- ssize_t [send_udp_pack](#) ([udp_pack_t](#) pack)

Function to send UDP package.
- void [destroy_udp_pack](#) ([udp_pack_t](#) pack)

Function free UDP package.

6.1.1 Detailed Description

Group function for work udp package.

6.1.2 Typedef Documentation

6.1.2.1 [udp_pack_t](#)

```
typedef struct udp\_pack* udp\_pack\_t
```

UDP packet descriptor.

The primary type for working with the library. All functions accept this pointer as an argument.

6.1.3 Function Documentation

6.1.3.1 add_byte_udp_pack()

```
ssize_t add_byte_udp_pack (  
    udp_pack_t pack,  
    const uint8_t byte)
```

Function addition byte in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
---------	-------------	-----------------------

in	byte	Addition byte in UDP package.
----	------	-------------------------------

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_date_udp_pack(pack, 'n');
set_date_udp_pack(pack, '\0');
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.3.2 add_date_udp_pack()

```
ssize_t add_date_udp_pack (
    udp_pack_t pack,
    void * data,
    uint16_t size)
```

Function addition data in UDP package.

Note

You must call `init_udp_pack` before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>data</i>	Addition data in UDP package.
in	<i>size</i>	Size addition data for UDP package.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
const char * const message = "Hello, world";
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
add_date_udp_pack(pack, message, strlen(message));
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.3.3 calculate_checksum_udp_pack()

```
void calculate_checksum_udp_pack (
    udp_pack_t pack) [static]
```

Function calculate checksum for ip header and UDP package.

Parameters

in, out	<i>pack</i>	UDP package for work.
---------	-------------	-----------------------

Note

This function is private. Not used outside [udp_lib/udp.c](#)

6.1.3.4 checksum_compute()

```
uint16_t checksum_compute (
    uint32_t sum) [static]
```

Function calculating from sum big endian to checksum big endian.

Parameters

in	<i>sum</i>	Sum in big endian.
----	------------	--------------------

Returns

Checksum in big endian.

Note

This function is private. Not used outside [udp_lib/udp.c](#)

6.1.3.5 destroy_udp_pack()

```
void destroy_udp_pack (
    udp_pack_t pack)
```

Function free UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work. Usage example.
---------	-------------	--------------------------------------

```

ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
// other code whit using udp_pack_t
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.3.6 get_pack_udp_pack()

```
void * get_pack_udp_pack (
    udp_pack_t pack) [static]
```

Function raw get pointer on UDP package.

Parameters

in, out	<i>pack</i>	UDP package for work.
---------	-------------	-----------------------

Note

This function is private. Not used outside [udp_lib/udp.c](#)

6.1.3.7 get_size_data_udp_pack()

```
uint16_t get_size_data_udp_pack (
    udp_pack_t pack)
```

Function give size data in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
---------	-------------	-----------------------

Returns

size data in UDP package. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
get_size_data_udp_pack(pack);
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.3.8 inet_port()

```
uint16_t inet_port (
    const char *const port) [static]
```

Function support for parsing port.

Parameters

in	<i>port</i>	Port in string format.
----	-------------	------------------------

Returns

Port for big indian architect.

Note

This function is private. Not used outside [udp_lib/udp.c](#)

6.1.3.9 init_udp_pack()

```
udp_pack_t init_udp_pack (
    void )
```

Function for create object UDP package.

Note

You must call [destroy_udp_pack](#) after this.

Returns

pointer or null on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
// other code whit using udp_pack_t
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.3.10 send_udp_pack()

```
ssize_t send_udp_pack (
    udp_pack_t pack)
```

Function to send UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
---------	-------------	-----------------------

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
send_udp_pack(pack);
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.3.11 set_data_udp_pack()

```
ssize_t set_data_udp_pack (
    udp_pack_t pack,
    void * data,
    const uint16_t size)
```

Function overriding data in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>data</i>	New data in UDP package.
in	<i>size</i>	Size new data for UDP package.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
const char * const message = "Hello, world";
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_data_udp_pack(pack, message, strlen(message));
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.3.12 set_file_data_udp_pack()

```
ssize_t set_file_data_udp_pack (
    udp_pack_t pack,
    const char *const file_name)
```

Function read data from file in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>file_name</i>	File name for read in UDP pack.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_file_data_udp_pack(pack, "Hello.txt");
get_not_udp_pack:
destroy_udp_pack(pack);
```


6.1.3.13 set_input_data_udp_pack()

```
ssize_t set_input_data_udp_pack (  
    udp_pack_t pack)
```

Function read data from stdin in UDP package.

Note

You must call `init_udp_pack` before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
---------	-------------	-----------------------

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;  
udp_pack_t pack = init_udp_pack();  
if (pack == NULL) {  
    ret = -1;  
    goto get_not_udp_pack;  
}  
set_input_data_udp_pack(pack);  
get_not_udp_pack:  
destroy_udp_pack(pack);
```

6.1.3.14 set_interface_udp_pack()

```
ssize_t set_interface_udp_pack (  
    udp_pack_t pack,  
    const char *const interface)
```

Function for pick interface to send UDP package.

Note

You must call `init_udp_pack` before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>interface</i>	Interface to send UDP pack.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;  
udp_pack_t pack = init_udp_pack();  
if (pack == NULL) {  
    ret = -1;  
    goto get_not_udp_pack;  
}  
set_interface_udp_pack(pack, "lo");  
get_not_udp_pack:  
destroy_udp_pack(pack);
```

6.1.3.15 set_ip_destination_udp_pack()

```
ssize_t set_ip_destination_udp_pack (  
    udp_pack_t pack,  
    const char *const ip)
```

Function for setting destination ip in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>ip</i>	Destination ip to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;  
udp_pack_t pack = init_udp_pack();  
if (pack == NULL) {  
    ret = -1;  
    goto get_not_udp_pack;  
}  
set_ip_destination_udp_pack(pack, "127.0.0.1");  
get_not_udp_pack:  
destroy_udp_pack(pack);
```

6.1.3.16 set_ip_source_udp_pack()

```
ssize_t set_ip_source_udp_pack (  
    udp_pack_t pack,  
    const char *const ip)
```

Function for setting source ip in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>ip</i>	Source ip to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;  
udp_pack_t pack = init_udp_pack();  
if (pack == NULL) {  
    ret = -1;  
    goto get_not_udp_pack;  
}  
set_ip_source_udp_pack(pack, "127.0.0.1");  
get_not_udp_pack:  
destroy_udp_pack(pack);
```

6.1.3.17 set_port_destination_udp_pack()

```
ssize_t set_port_destination_udp_pack (  
    udp_pack_t pack,  
    const char *const port)
```

Function for setting destination port in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>port</i>	Destination port to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;  
udp_pack_t pack = init_udp_pack();  
if (pack == NULL) {  
    ret = -1;  
    goto get_not_udp_pack;  
}  
set_port_destination_udp_pack(pack, "8003");  
get_not_udp_pack:  
destroy_udp_pack(pack);
```

6.1.3.18 set_port_source_udp_pack()

```
ssize_t set_port_source_udp_pack (  
    udp_pack_t pack,  
    const char *const port)
```

Function for setting source port in UDP package.

Note

You must call [init_udp_pack](#) before this.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>port</i>	Source port to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;  
udp_pack_t pack = init_udp_pack();  
if (pack == NULL) {  
    ret = -1;  
    goto get_not_udp_pack;  
}  
set_port_source_udp_pack(pack, "8001");  
get_not_udp_pack:  
destroy_udp_pack(pack);
```

6.1.3.19 set_size_udp_pack()

```
void set_size_udp_pack (  
    udp_pack_t pack,  
    const uint16_t size) [static]
```

Function raw write size UDP package.

Parameters

in, out	<i>pack</i>	UDP package for work.
in	<i>size</i>	New size UDP package.

Note

This function is private. Not used outside [udp_lib/udp.c](#)

6.1.3.20 sum_compute()

```
uint32_t sum_compute (  
    void * ptr,  
    uint16_t nbytes) [static]
```

Function calculate sum in big endian.

Parameters

in	<i>ptr</i>	Buffer for calculating sum in big endian.
in	<i>nbytes</i>	Size buffer for calculating sum.

Returns

Sum buffer in big endian.

Note

This function is private. Not used outside [udp_lib/udp.c](#)

Chapter 7

Directory Documentation

7.1 udp_lib Directory Reference

Files

- file [udp.c](#)
Code file for work udp package.
- file [udp.h](#)
Header file for work udp package.

Chapter 8

Class Documentation

8.1 `pseudo_header` Struct Reference

Struct `pseudo_header` for calculate checksum for UDP package.

Public Attributes

- `uint32_t` `source_address`
- `uint32_t` `dest_address`
- `uint8_t` `placeholder`
- `uint8_t` `protocol`
- `uint16_t` `udp_length`

8.1.1 Detailed Description

Struct `pseudo_header` for calculate checksum for UDP package.

Note

This struct is private. Not used outside `udp_lib/udp.c`

8.1.2 Member Data Documentation

8.1.2.1 `dest_address`

```
uint32_t pseudo_header::dest_address
```

Destination ip addresses

8.1.2.2 `placeholder`

```
uint8_t pseudo_header::placeholder
```

Placeholder is 0x00.

8.1.2.3 protocol

```
uint8_t pseudo_header::protocol
```

Constant protocol.

8.1.2.4 source_address

```
uint32_t pseudo_header::source_address
```

Source ip addresses

8.1.2.5 udp_length

```
uint16_t pseudo_header::udp_length
```

Length UDP package.

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

- [udp_lib/udp.c](#)

8.2 udp_head Struct Reference

Struct is header UDP pack.

Public Attributes

- uint16_t [m_port_source](#)
- uint16_t [m_port_destination](#)
- uint16_t [m_length](#)
- uint16_t [m_checksum](#)

8.2.1 Detailed Description

Struct is header UDP pack.

Note

This struct is private. Not used outside [udp_lib/udp.c](#)

8.2.2 Member Data Documentation

8.2.2.1 m_checksum

```
uint16_t udp_head::m_checksum
```

Calculated software checksum

8.2.2.2 m_length

```
uint16_t udp_head::m_length
```

Length udp pack

8.2.2.3 m_port_destination

```
uint16_t udp_head::m_port_destination
```

Port destination

8.2.2.4 m_port_source

```
uint16_t udp_head::m_port_source
```

Port source

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

- [udp_lib/udp.c](#)

8.3 udp_pack Struct Reference

Struct is UDP package.

Public Attributes

- `char * m_interface`
- `struct ethhdr m_ethhdr`
- `struct iphdr m_iphdr`
- `struct udp_head m_head`
- `uint8_t m_data [MAX_SIZE_DATA]`

8.3.1 Detailed Description

Struct is UDP package.

Note

This struct is private. Not used outside [udp_lib/udp.c](#)

8.3.2 Member Data Documentation

8.3.2.1 m_data

```
uint8_t udp_pack::m_data[MAX_SIZE_DATA]
```

Data in UDP package.

8.3.2.2 m_ethhdr

```
struct ethhdr udp_pack::m_ethhdr
```

Ethernet header start UDP package.

8.3.2.3 m_head

```
struct udp\_head udp_pack::m_head
```

UDP header

8.3.2.4 m_interface

```
char* udp_pack::m_interface
```

Name ethernet interface.

8.3.2.5 m_iphdr

```
struct iphdr udp_pack::m_iphdr
```

IP header

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

- [udp_lib/udp.c](#)

Chapter 9

File Documentation

9.1 main.c File Reference

```
#include "udp_lib/udp.h"
#include <getopt.h>
#include <stddef.h>
#include <string.h>
```

Functions

- int [main](#) (int argc, char **argv)

9.1.1 Function Documentation

9.1.1.1 main()

```
int main (
    int argc,
    char ** argv)
```

9.2 README.md File Reference

9.3 udp_lib/udp.c File Reference

Code file for work udp package.

```
#include "udp_lib/udp.h"
#include <stdint.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
```

```
#include <sys/ioctl.h>
#include <sys/stat.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <net/if.h>
#include <linux/if_packet.h>
#include <net/ethernet.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/ip.h>
```

Classes

- struct [udp_head](#)
Struct is header UDP pack.
- struct [udp_pack](#)
Struct is UDP package.
- struct [pseudo_header](#)
Struct [pseudo_header](#) for calculate checksum for UDP package.

Macros

- #define [PACKED](#) __attribute__((packed))
- #define [HEAD_ETH](#) sizeof(struct ethhdr)
- #define [HEAD_UDP](#) sizeof(struct [udp_head](#))
- #define [HEAD_IP](#) sizeof(struct iphdr)
- #define [HEAD_UDP_IP](#) ([HEAD_IP](#) + [HEAD_UDP](#))
- #define [MAX_SIZE_DATA](#) (0xFFFF - [HEAD_UDP_IP](#))
- #define [NULL_CHECKSUM](#) 0x0000
- #define [MIN](#)(left, right)
- #define [HEAD_PSEUDO](#) sizeof(struct [pseudo_header](#))

Functions

- [udp_pack_t](#) [init_udp_pack](#) (void)
Function for create object UDP package.
- static uint16_t [inet_port](#) (const char *const port)
Function support for parsing port.
- ssize_t [set_port_source_udp_pack](#) ([udp_pack_t](#) pack, const char *const port)
Function for setting source port in UDP package.
- ssize_t [set_port_destination_udp_pack](#) ([udp_pack_t](#) pack, const char *const port)
Function for setting destination port in UDP package.
- ssize_t [set_ip_source_udp_pack](#) ([udp_pack_t](#) pack, const char *const ip)
Function for setting source ip in UDP package.
- ssize_t [set_ip_destination_udp_pack](#) ([udp_pack_t](#) pack, const char *const ip)
Function for setting destination ip in UDP package.
- static void [set_size_udp_pack](#) ([udp_pack_t](#) pack, const uint16_t size)
Function raw write size UDP package.
- uint16_t [get_size_data_udp_pack](#) ([udp_pack_t](#) pack)

- Function give size data in UDP package.*
- ssize_t [add_date_udp_pack](#) ([udp_pack_t](#) pack, void *data, const uint16_t size)
- Function addition data in UDP package.*
- ssize_t [set_data_udp_pack](#) ([udp_pack_t](#) pack, void *data, uint16_t size)
- Function overriding data in UDP package.*
- ssize_t [add_byte_udp_pack](#) ([udp_pack_t](#) pack, uint8_t byte)
- Function addition byte in UDP package.*
- ssize_t [set_input_data_udp_pack](#) ([udp_pack_t](#) pack)
- Function read data from stdin in UDP package.*
- ssize_t [set_file_data_udp_pack](#) ([udp_pack_t](#) pack, const char *const file_name)
- Function read data from file in UDP package.*
- static uint32_t [sum_compute](#) (void *ptr, uint16_t nbytes)
- Function calculate sum in big endian.*
- static uint16_t [checksum_compute](#) (uint32_t sum)
- Function calculating from sum big endian to checksum big endian.*
- static void [calculate_checksum_udp_pack](#) ([udp_pack_t](#) pack)
- Function calculate checksum for ip header and UDP package.*
- ssize_t [set_interface_udp_pack](#) ([udp_pack_t](#) pack, const char *const interface)
- Function for pick interface to send UDP package.*
- static void * [get_pack_udp_pack](#) ([udp_pack_t](#) pack)
- Function raw get pointer on UDP package.*
- ssize_t [send_udp_pack](#) ([udp_pack_t](#) pack)
- Function to send UDP package.*
- void [destroy_udp_pack](#) ([udp_pack_t](#) pack)
- Function free UDP package.*

9.3.1 Detailed Description

Code file for work udp package.

Author

Vladsanin777

9.3.2 Macro Definition Documentation

9.3.2.1 HEAD_ETH

```
#define HEAD_ETH sizeof(struct ethhdr)
```

9.3.2.2 HEAD_IP

```
#define HEAD_IP sizeof(struct iphdr)
```

9.3.2.3 HEAD_PSEUDO

```
#define HEAD_PSEUDO sizeof(struct pseudo\_header)
```

9.3.2.4 HEAD_UDP

```
#define HEAD_UDP sizeof(struct udp_head)
```

9.3.2.5 HEAD_UDP_IP

```
#define HEAD_UDP_IP (HEAD_IP + HEAD_UDP)
```

9.3.2.6 MAX_SIZE_DATA

```
#define MAX_SIZE_DATA (0xFFFF - HEAD_UDP_IP)
```

9.3.2.7 MIN

```
#define MIN(  
    left,  
    righth)
```

Value:

```
((left) < (righth)) ? (left) : (righth)
```

9.3.2.8 NULL_CHECKSUM

```
#define NULL_CHECKSUM 0x0000
```

9.3.2.9 PACKED

```
#define PACKED __attribute__((packed))
```

9.4 udp_lib/udp.h File Reference

Header file for work udp package.

```
#include <stdint.h>  
#include <stdlib.h>
```

Typedefs

- typedef struct [udp_pack](#) * [udp_pack_t](#)
UDP packet descriptor.

Functions

- `udp_pack_t init_udp_pack` (void)
Function for create object UDP package.
- `ssize_t set_port_source_udp_pack` (udp_pack_t pack, const char *const port)
Function for setting source port in UDP package.
- `ssize_t set_port_destination_udp_pack` (udp_pack_t pack, const char *const port)
Function for setting destination port in UDP package.
- `ssize_t set_ip_source_udp_pack` (udp_pack_t pack, const char *const ip)
Function for setting source ip in UDP package.
- `ssize_t set_ip_destination_udp_pack` (udp_pack_t pack, const char *const ip)
Function for setting destination ip in UDP package.
- `uint16_t get_size_data_udp_pack` (udp_pack_t pack)
Function give size data in UDP package.
- `ssize_t add_data_udp_pack` (udp_pack_t pack, void *data, uint16_t size)
Function addition data in UDP package.
- `ssize_t set_data_udp_pack` (udp_pack_t pack, void *data, const uint16_t size)
Function overriding data in UDP package.
- `ssize_t add_byte_udp_pack` (udp_pack_t pack, const uint8_t byte)
Function addition byte in UDP package.
- `ssize_t set_input_data_udp_pack` (udp_pack_t pack)
Function read data from stdin in UDP package.
- `ssize_t set_file_data_udp_pack` (udp_pack_t pack, const char *const file_name)
Function read data from file in UDP package.
- `ssize_t set_interface_udp_pack` (udp_pack_t pack, const char *const interface)
Function for pick interface to send UDP package.
- `ssize_t send_udp_pack` (udp_pack_t pack)
Function to send UDP package.
- `void destroy_udp_pack` (udp_pack_t pack)
Function free UDP package.

9.4.1 Detailed Description

Header file for work udp package.

Author

Vladsanin777

9.5 udp.h

[Go to the documentation of this file.](#)

```
00001
00006
00007 #include <stdint.h>
00008 #include <stdlib.h>
00009
00015
00019 struct udp_pack;
00020
00026 typedef struct udp_pack * udp_pack_t;
00027
00045 udp_pack_t init_udp_pack(void);
```

```
00046
00066 ssize_t set_port_source_udp_pack(udp_pack_t pack, const char * const port);
00067
00087 ssize_t set_port_destination_udp_pack(udp_pack_t pack, const char * const port);
00088
00108 ssize_t set_ip_source_udp_pack(udp_pack_t pack, const char * const ip);
00109
00129 ssize_t set_ip_destination_udp_pack(udp_pack_t pack, const char * const ip);
00130
00149 uint16_t get_size_data_udp_pack(udp_pack_t pack);
00150
00172 ssize_t add_data_udp_pack(udp_pack_t pack, void * data, uint16_t size);
00173
00195 ssize_t set_data_udp_pack(udp_pack_t pack, void * data, \
00196     const uint16_t size);
00197
00218 ssize_t add_byte_udp_pack(udp_pack_t pack, \
00219     const uint8_t byte);
00220
00239 ssize_t set_input_data_udp_pack(udp_pack_t pack);
00240
00260 ssize_t set_file_data_udp_pack( \
00261     udp_pack_t pack, const char * const file_name);
00262
00282 ssize_t set_interface_udp_pack( \
00283     udp_pack_t pack, const char * const interface);
00284
00303 ssize_t send_udp_pack(udp_pack_t pack);
00304
00322 void destroy_udp_pack(udp_pack_t pack);
00323
```


Index

- add_byte_udp_pack
 - work for udp, [13](#)
- add_date_udp_pack
 - work for udp, [14](#)
- calculate_checksum_udp_pack
 - work for udp, [14](#)
- checksum_compute
 - work for udp, [15](#)
- dest_address
 - pseudo_header, [25](#)
- destroy_udp_pack
 - work for udp, [15](#)
- get_pack_udp_pack
 - work for udp, [15](#)
- get_size_data_udp_pack
 - work for udp, [16](#)
- HEAD_ETH
 - udp.c, [31](#)
- HEAD_IP
 - udp.c, [31](#)
- HEAD_PSEUDO
 - udp.c, [31](#)
- HEAD_UDP
 - udp.c, [31](#)
- HEAD_UDP_IP
 - udp.c, [32](#)
- inet_port
 - work for udp, [16](#)
- init_udp_pack
 - work for udp, [17](#)
- m_checksum
 - udp_head, [27](#)
- m_data
 - udp_pack, [28](#)
- m_ethhdr
 - udp_pack, [28](#)
- m_head
 - udp_pack, [28](#)
- m_interface
 - udp_pack, [28](#)
- m_iphdr
 - udp_pack, [28](#)
- m_length
 - udp_head, [27](#)
- m_port_destination
 - udp_head, [27](#)
- m_port_source
 - udp_head, [27](#)
- main
 - main.c, [29](#)
- main.c, [29](#)
 - main, [29](#)
- MAX_SIZE_DATA
 - udp.c, [32](#)
- MIN
 - udp.c, [32](#)
- NULL_CHECKSUM
 - udp.c, [32](#)
- PACKED
 - udp.c, [32](#)
- placeholder
 - pseudo_header, [25](#)
- protocol
 - pseudo_header, [25](#)
- pseudo_header, [25](#)
 - dest_address, [25](#)
 - placeholder, [25](#)
 - protocol, [25](#)
 - source_address, [26](#)
 - udp_length, [26](#)
- README.md, [29](#)
- send_udp_pack
 - work for udp, [17](#)
- set_data_udp_pack
 - work for udp, [17](#)
- set_file_data_udp_pack
 - work for udp, [18](#)
- set_input_data_udp_pack
 - work for udp, [18](#)
- set_interface_udp_pack
 - work for udp, [19](#)
- set_ip_destination_udp_pack
 - work for udp, [19](#)
- set_ip_source_udp_pack
 - work for udp, [20](#)
- set_port_destination_udp_pack
 - work for udp, [20](#)
- set_port_source_udp_pack
 - work for udp, [21](#)
- set_size_udp_pack
 - work for udp, [21](#)

- source_address
 - pseudo_header, [26](#)
- sum_compute
 - work for udp, [22](#)
- udp, [1](#)
- udp.c
 - HEAD_ETH, [31](#)
 - HEAD_IP, [31](#)
 - HEAD_PSEUDO, [31](#)
 - HEAD_UDP, [31](#)
 - HEAD_UDP_IP, [32](#)
 - MAX_SIZE_DATA, [32](#)
 - MIN, [32](#)
 - NULL_CHECKSUM, [32](#)
 - PACKED, [32](#)
- udp_head, [26](#)
 - m_checksum, [27](#)
 - m_length, [27](#)
 - m_port_destination, [27](#)
 - m_port_source, [27](#)
- udp_length
 - pseudo_header, [26](#)
- udp_lib Directory Reference, [23](#)
- udp_lib/udp.c, [29](#)
- udp_lib/udp.h, [32](#), [33](#)
- udp_pack, [27](#)
 - m_data, [28](#)
 - m_ethhdr, [28](#)
 - m_head, [28](#)
 - m_interface, [28](#)
 - m_iphdr, [28](#)
- udp_pack_t
 - work for udp, [12](#)
- work for udp, [11](#)
 - add_byte_udp_pack, [13](#)
 - add_data_udp_pack, [14](#)
 - calculate_checksum_udp_pack, [14](#)
 - checksum_compute, [15](#)
 - destroy_udp_pack, [15](#)
 - get_pack_udp_pack, [15](#)
 - get_size_data_udp_pack, [16](#)
 - inet_port, [16](#)
 - init_udp_pack, [17](#)
 - send_udp_pack, [17](#)
 - set_data_udp_pack, [17](#)
 - set_file_data_udp_pack, [18](#)
 - set_input_data_udp_pack, [18](#)
 - set_interface_udp_pack, [19](#)
 - set_ip_destination_udp_pack, [19](#)
 - set_ip_source_udp_pack, [20](#)
 - set_port_destination_udp_pack, [20](#)
 - set_port_source_udp_pack, [21](#)
 - set_size_udp_pack, [21](#)
 - sum_compute, [22](#)
 - udp_pack_t, [12](#)