

UDP

Generated by Doxygen 1.16.1



---

<b>1 udp</b>	<b>1</b>
<b>2 Topic Index</b>	<b>3</b>
2.1 Topics . . . . .	3
<b>3 Directory Hierarchy</b>	<b>5</b>
3.1 Directories . . . . .	5
<b>4 Class Index</b>	<b>7</b>
4.1 Class List . . . . .	7
<b>5 File Index</b>	<b>9</b>
5.1 File List . . . . .	9
<b>6 Topic Documentation</b>	<b>11</b>
6.1 work for udp . . . . .	11
6.1.1 Detailed Description . . . . .	13
6.1.2 Macro Definition Documentation . . . . .	13
6.1.2.1 PACKED . . . . .	13
6.1.3 Typedef Documentation . . . . .	13
6.1.3.1 udp_pack_t . . . . .	13
6.1.4 Function Documentation . . . . .	13
6.1.4.1 add_byte_udp_pack() . . . . .	13
6.1.4.2 add_date_udp_pack() . . . . .	14
6.1.4.3 calculate_checksum_udp_pack() . . . . .	15
6.1.4.4 checksum_compute() . . . . .	15
6.1.4.5 destroy_udp_pack() . . . . .	15
6.1.4.6 get_data_hex_udp_pack() . . . . .	16
6.1.4.7 get_data_udp_pack() . . . . .	16
6.1.4.8 get_interface_udp_pack() . . . . .	17
6.1.4.9 get_ip_address_destantion_udp_pack() . . . . .	18
6.1.4.10 get_ip_address_source_udp_pack() . . . . .	18
6.1.4.11 get_mac_address_destantion_udp_pack() . . . . .	19
6.1.4.12 get_mac_address_source_udp_pack() . . . . .	20
6.1.4.13 get_pack_udp_pack() . . . . .	20
6.1.4.14 get_port_destantion_udp_pack() . . . . .	21
6.1.4.15 get_port_source_udp_pack() . . . . .	21
6.1.4.16 get_size_data_udp_pack() . . . . .	22
6.1.4.17 inet_mac() . . . . .	23
6.1.4.18 inet_port() . . . . .	24
6.1.4.19 init_udp_pack() . . . . .	24
6.1.4.20 print_udp_pack() . . . . .	25
6.1.4.21 send_udp_pack() . . . . .	25
6.1.4.22 set_data_udp_pack() . . . . .	26

6.1.4.23 set_file_data_udp_pack() . . . . .	26
6.1.4.24 set_input_data_udp_pack() . . . . .	27
6.1.4.25 set_interface_udp_pack() . . . . .	27
6.1.4.26 set_ip_address_destantion_udp_pack()	28
6.1.4.27 set_ip_address_source_udp_pack() . .	28
6.1.4.28 set_mac_address_destantion_udp_pack()	29
6.1.4.29 set_mac_address_source_udp_pack() . .	29
6.1.4.30 set_port_destantion_udp_pack() . . . .	30
6.1.4.31 set_port_source_udp_pack() . . . . .	30
6.1.4.32 set_size_udp_pack() . . . . .	31
6.1.4.33 sum_compute() . . . . .	31
6.1.5 Variable Documentation . . . . .	32
6.1.5.1 PACKED . . . . .	32
<b>7 Directory Documentation</b>	<b>33</b>
7.1 udp_lib Directory Reference . . . . .	33
<b>8 Class Documentation</b>	<b>35</b>
8.1 pseudo_header Struct Reference . . . . .	35
8.1.1 Detailed Description . . . . .	35
8.1.2 Member Data Documentation . . . . .	35
8.1.2.1 dest_address . . . . .	35
8.1.2.2 placeholder . . . . .	35
8.1.2.3 protocol . . . . .	36
8.1.2.4 source_address . . . . .	36
8.1.2.5 udp_length . . . . .	36
8.2 udp_head Struct Reference . . . . .	36
8.2.1 Detailed Description . . . . .	36
8.2.2 Member Data Documentation . . . . .	37
8.2.2.1 m_checksum . . . . .	37
8.2.2.2 m_length . . . . .	37
8.2.2.3 m_port_destantion . . . . .	37
8.2.2.4 m_port_source . . . . .	37
8.3 udp_pack Struct Reference . . . . .	37
8.3.1 Detailed Description . . . . .	38
8.3.2 Member Data Documentation . . . . .	38
8.3.2.1 m_data . . . . .	38
8.3.2.2 m_ethhdr . . . . .	38
8.3.2.3 m_head . . . . .	38
8.3.2.4 m_interface . . . . .	38
8.3.2.5 m_iphdr . . . . .	38
<b>9 File Documentation</b>	<b>39</b>

---

9.1 main.c File Reference . . . . .	39
9.1.1 Function Documentation . . . . .	39
9.1.1.1 main() . . . . .	39
9.2 README.md File Reference . . . . .	39
9.3 udp_lib/udp.c File Reference . . . . .	39
9.3.1 Detailed Description . . . . .	42
9.3.2 Macro Definition Documentation . . . . .	42
9.3.2.1 HEAD_ETH . . . . .	42
9.3.2.2 HEAD_IP . . . . .	42
9.3.2.3 HEAD_PSEUDO . . . . .	42
9.3.2.4 HEAD_UDP . . . . .	42
9.3.2.5 HEAD_UDP_IP . . . . .	42
9.3.2.6 MAX_SIZE_DATA . . . . .	42
9.3.2.7 MIN . . . . .	42
9.3.2.8 NULL_CHECKSUM . . . . .	43
9.4 udp_lib/udp.h File Reference . . . . .	43
9.4.1 Detailed Description . . . . .	44
9.5 udp.h . . . . .	44

**Index**

47



# **Chapter 1**

**udp**



# **Chapter 2**

## **Topic Index**

### **2.1 Topics**

Here is a list of all topics with brief descriptions:

work for udp . . . . .	<a href="#">11</a>
------------------------	--------------------



# Chapter 3

## Directory Hierarchy

### 3.1 Directories

udp_lib	33
udp.c	39
udp.h	43



# Chapter 4

## Class Index

### 4.1 Class List

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

<a href="#">pseudo_header</a>	Struct <code>pseudo_header</code> for calculate checksum for UDP package . . . . .	35
<a href="#">udp_head</a>	Struct is header UDP pack . . . . .	36
<a href="#">udp_pack</a>	Struct is UDP package . . . . .	37



# Chapter 5

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.c</a> . . . . .	39
<a href="#">udp_lib/udp.c</a>	
Code file for work udp package . . . . .	39
<a href="#">udp_lib/udp.h</a>	
Header file for work udp package . . . . .	43



# 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.*

#### Macros

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

#### 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.*
- static void \* [inet\\_mac \(const char \\*const mac\\_address\)](#)  
*Function return mac address in big endian.*
- [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\\_destantion\\_udp\\_pack \(udp\\_pack\\_t pack, const char \\*const port\)](#)  
*Function for setting destantion port in UDP package.*
- ssize\_t [set\\_ip\\_address\\_source\\_udp\\_pack \(udp\\_pack\\_t pack, const char \\*const ip\)](#)  
*Function for setting source ip address in UDP package.*
- ssize\_t [set\\_ip\\_address\\_destantion\\_udp\\_pack \(udp\\_pack\\_t pack, const char \\*const ip\)](#)  
*Function for setting destantion ip address in 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 \(udp\\_pack\\_t pack, void \\*data, uint16\\_t size\)](#)  
*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.*
- char \* [get\\_mac\\_address\\_source\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)  
*Function getting mac address for source.*
- char \* [get\\_mac\\_address\\_destantion\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)  
*Function setting mac address for destantion.*
- ssize\_t [set\\_mac\\_address\\_source\\_udp\\_pack \(udp\\_pack\\_t pack, const char \\*const mac\\_address\)](#)  
*Function setting mac address for source.*
- ssize\_t [set\\_mac\\_address\\_destantion\\_udp\\_pack \(udp\\_pack\\_t pack, const char \\*const mac\\_address\)](#)  
*Function setting mac address for destantion.*
- char \* [get\\_interface\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)  
*Function for getting interface.*
- char \* [get\\_port\\_source\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)  
*Function for getting port source.*
- char \* [get\\_port\\_destantion\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)  
*Function for getting port destantion.*
- char \* [get\\_ip\\_address\\_source\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)  
*Function for getting ip address source.*
- char \* [get\\_ip\\_address\\_destantion\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)  
*Function for getting ip address destantion.*
- char \* [get\\_data\\_udp\\_pack \(udp\\_pack\\_t pack\)](#)

- `char * get_data_hex_udp_pack (udp_pack_t pack)`  
*Function for getting data.*
- `ssize_t print_udp_pack (udp_pack_t pack)`  
*Function for getting data hex.*
- `void destroy_udp_pack (udp_pack_t pack)`  
*Function print all info about udp pack.*
- `void destroy_udp_pack (udp_pack_t pack)`  
*Function free UDP package.*

## Variables

- `struct udp_head PACKED`

### 6.1.1 Detailed Description

Group function for work udp package.

### 6.1.2 Macro Definition Documentation

#### 6.1.2.1 PACKED

```
struct pseudo_header PACKED __attribute__((packed))
```

### 6.1.3 Typedef Documentation

#### 6.1.3.1 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.

### 6.1.4 Function Documentation

#### 6.1.4.1 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	pack	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);
```

**6.1.4.2 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);
```

#### 6.1.4.3 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](#)

#### 6.1.4.4 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](#)

#### 6.1.4.5 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. <pre>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);</pre>
---------	-------------	---

#### 6.1.4.6 `get_data_hex_udp_pack()`

```
char * get_data_hex_udp_pack (
    udp_pack_t pack)
```

Function for getting data hex.

##### Note

You must call `init_udp_pack` before this.

##### Parameters

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

##### Returns

Data hex in UDP package or NULL pointer 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;
}
{
    char * data = get_data_hex_udp_pack(pack);
    if (data == NULL) {
        ret = -1;
        goto get_not_data_hex_udp_pack;
    }
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
get_not_data_hex_udp_pack:
get_not_udp_pack:
destroy_udp_pack(pack);
```

#### 6.1.4.7 `get_data_udp_pack()`

```
char * get_data_udp_pack (
    udp_pack_t pack)
```

Function for getting data.

##### Note

You must call `init_udp_pack` before this.

##### Parameters

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

**Returns**

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;
}
void * data = get_data_udp_pack(pack);
// other code whit udp_pack_t
}
// other code whit udp_pack_t
get_not_udp_pack:
destroy_udp_pack(pack);
```

**6.1.4.8 get\_interface\_udp\_pack()**

```
char * get_interface_udp_pack (
    udp_pack_t pack)
```

Function for getting interface.

**Note**

You must call [init\\_udp\\_pack](#) before this.

You can call [set\\_interface\\_udp\\_pack](#) before this.

**Parameters**

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

**Returns**

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_interface_udp_pack(pack, "lo");
if (ret == -1)
    goto setting_not_interface;
{
    char * interface = NULL;
    interface = get_interface_udp_pack(pack);
    if (interface == NULL)
        getting_not_interface;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_interface:
setting_not_interface:
get_not_udp_pack:
destroy_udp_pack(pack);
```

### 6.1.4.9 get\_ip\_address\_destantion\_udp\_pack()

```
char * get_ip_address_destantion_udp_pack (
    udp_pack_t pack)
```

Function for getting ip address destantion.

#### Note

You must call [init\\_udp\\_pack](#) before this.

You can call [set\\_ip\\_address\\_destantion\\_udp\\_pack](#) before this.

#### Parameters

in, out	pack	UDP package for work.
---------	------	-----------------------

#### Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_ip_address_source_udp_pack(pack, "127.0.0.1");
if (ret == -1)
    goto setting_not_ip_destantion;
{
    char * ip_address = NULL;
    ip_address = get_ip_destantion_udp_pack(pack);
    if (ip_address == NULL)
        getting_not_ip_address;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_ip_address:
setting_not_ip_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

### 6.1.4.10 get\_ip\_address\_source\_udp\_pack()

```
char * get_ip_address_source_udp_pack (
    udp_pack_t pack)
```

Function for getting ip address source.

#### Note

You must call [init\\_udp\\_pack](#) before this.

You can call [set\\_ip\\_address\\_source\\_udp\\_pack](#) before this.

#### Parameters

in, out	pack	UDP package for work.
---------	------	-----------------------

**Returns**

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init\_udp\_pack\(\);
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set\_ip\_address\_source\_udp\_pack(pack, "127.0.0.1");
if (ret == -1)
    goto setting_not_ip_address;
{
    char * ip_address = NULL;
    ip_address = get\_ip\_address\_source\_udp\_pack(pack);
    if (ip_source == NULL)
        getting_not_ip_address;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_ip_address:
setting_not_ip_address:
get_not_udp_pack:
destroy\_udp\_pack(pack);
```

**6.1.4.11 [get\\_mac\\_address\\_destantion\\_udp\\_pack\(\)](#)**

```
char * get\_mac\_address\_destantion\_udp\_pack (
    udp_pack_t pack)
```

Function setting mac address for destantion.

**Note**

You must call [init\\_udp\\_pack](#) before this.

You can call [set\\_mac\\_address\\_destantion\\_udp\\_pack](#) before this.

**Parameters**

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

**Returns**

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init\_udp\_pack\(\);
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set\_mac\_address\_destantion\_udp\_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
{
    char * mac_address = NULL;
    mac_address = get\_mac\_address\_destantion\_udp\_pack(udp_pack_t pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy\_udp\_pack(pack);
```

#### 6.1.4.12 get\_mac\_address\_source\_udp\_pack()

```
char * get_mac_address_source_udp_pack (
    udp_pack_t pack)
```

Function getting mac address for source.

##### Note

You must call [init\\_udp\\_pack](#) before this.

You can call [set\\_mac\\_address\\_source\\_udp\\_pack](#) before this.

##### Parameters

in, out	pack	UDP package for work.
---------	------	-----------------------

##### Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_mac_address_source_udp_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
{
    char * mac_address = NULL;
    mac_address = get_mac_address_source_udp_pack(pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

#### 6.1.4.13 get\_pack\_udp\_pack()

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

Function raw get pointer on UDP package.

##### Parameters

in, out	pack	UDP package for work.
---------	------	-----------------------

##### Returns

Getting UDP package.

##### Note

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

#### 6.1.4.14 get\_port\_destantion\_udp\_pack()

```
char * get_port_destantion_udp_pack (
    udp_pack_t pack)
```

Function for getting port destantion.

##### Note

You must call [init\\_udp\\_pack](#) before this.

You can call [set\\_port\\_destantion\\_udp\\_pack](#) before this.

##### Parameters

in, out	pack	UDP package for work.
---------	------	-----------------------

##### Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_port_destantion_udp_pack(pack, "8001");
if (ret == -1)
    goto setting_not_port_destantion;
{
    char * port_destantion = NULL;
    port_destantion = get_port_destantion_udp_pack(pack);
    if (port_destantion == NULL)
        getting_not_port_destantion;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_port_destantion:
setting_not_port_destantion:
get_not_udp_pack:
destroy_udp_pack(pack);
```

#### 6.1.4.15 get\_port\_source\_udp\_pack()

```
char * get_port_source_udp_pack (
    udp_pack_t pack)
```

Function for getting port source.

##### Note

You must call [init\\_udp\\_pack](#) before this.

You can call [set\\_port\\_source\\_udp\\_pack](#) before this.

##### Parameters

in, out	pack	UDP package for work.
---------	------	-----------------------

**Returns**

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_port_source_udp_pack(pack, "8003");
if (ret == -1)
    goto setting_not_port_source;
{
    char * port_source = NULL;
    port_source = get_port_source_udp_pack(pack);
    if (port_source == NULL)
        getting_not_port_source;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_port_source:
setting_not_port_source:
get_not_udp_pack:
destroy_udp_pack(pack);
```

**6.1.4.16 `get_size_data_udp_pack()`**

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

Function give size data in UDP package.

Function for getting size data.

**Note**

You must call `init_udp_pack` before this.

**Parameters**

<code>in, out</code>	<code>pack</code>	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);
```

**Note**

You must call `init_udp_pack` before this.

You can call `set_data_udp_pack` before this.

You can call `add_data_udp_pack` before this.

You can call `add_symbol_udp_pack` before this.

**Parameters**

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

**Returns**

Length 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;
}
{
    uint16_t size_data = get_size_data_udp_pack(pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_size_data:
get_not_udp_pack:
destroy_udp_pack(pack);
```

Function give size data in UDP package.

**Note**

- You must call `init_udp_pack` before this.
- You can call `set_data_udp_pack` before this.
- You can call `add_data_udp_pack` before this.
- You can call `add_symbol_udp_pack` before this.

**Parameters**

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

**Returns**

Length 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;
}
{
    uint16_t size_data = get_size_data_udp_pack(pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_size_data:
get_not_udp_pack:
destroy_udp_pack(pack);
```

**6.1.4.17 inet\_mac()**

```
void * inet_mac (
    const char *const mac_address) [static]
```

Function return mac address in big endian.

**Parameters**

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

**Returns**

Getting buffer mac address.

**Note**

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

**6.1.4.18 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](#)

**6.1.4.19 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);
```

#### 6.1.4.20 print\_udp\_pack()

```
ssize_t print_udp_pack (
    udp_pack_t pack)
```

Function print all info about udp pack.

##### Note

You must call [init\\_udp\\_pack](#) before this.

##### Parameters

in, out	pack	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;
}
ret = print_udp_pack(pack);
if (ret)
    goto print_not_udp_pack;
// other code whit udp_pack_t
print_not_udp_pack:
get_not_udp_pack:
destroy_udp_pack(pack);
```

#### 6.1.4.21 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	pack	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);
```

#### 6.1.4.22 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_date_udp_pack(pack, message, strlen(message));
get_not_udp_pack:
destroy_udp_pack(pack);
```

#### 6.1.4.23 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();
```

**6.1.4.24 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);
```

**6.1.4.25 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);
```

#### 6.1.4.26 set\_ip\_address\_destantion\_udp\_pack()

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

Function for setting destantion ip address 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 address 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;
}
ret = set_ip_address_destantion_udp_pack(pack, "127.0.0.1");
if (ret == -1)
    goto setting_not_ip_address;
// other code whit udp_pack_t
setting_not_ip_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

#### 6.1.4.27 set\_ip\_address\_source\_udp\_pack()

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

Function for setting source ip address 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;
}
ret = set_ip_address_source_udp_pack(pack, "127.0.0.1");
if (ret == -1)
    goto set_not_ip_address;
// other code whit udp_pack_t
set_not_ip_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

**6.1.4.28 set\_mac\_address\_destantion\_udp\_pack()**

```
ssize_t set_mac_address_destantion_udp_pack (
    udp_pack_t pack,
    const char *const mac_address)
```

Function setting mac address for destantion.

**Note**

You must call [init\\_udp\\_pack](#) before this.

**Parameters**

in, out	<i>pack</i>	UDP package for work.
in	<i>mac_address</i>	String is mac address destantion.

**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;
}
ret = set_mac_address_destantion_udp_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

**6.1.4.29 set\_mac\_address\_source\_udp\_pack()**

```
ssize_t set_mac_address_source_udp_pack (
    udp_pack_t pack,
    const char *const mac_address)
```

Function setting mac address for source.

**Note**

You must call [init\\_udp\\_pack](#) before this.

**Parameters**

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

in	<i>mac_address</i>	String is mac address source.
----	--------------------	-------------------------------

**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;
}
ret = set_mac_address_source_udp_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

**6.1.4.30 set\_port\_destantion\_udp\_pack()**

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

Function for setting destantion 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_destantion_udp_pack(pack, "8003");
get_not_udp_pack:
destroy_udp_pack(pack);
```

**6.1.4.31 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);
```

**6.1.4.32 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](#)

**6.1.4.33 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](#)

## 6.1.5 Variable Documentation

### 6.1.5.1 PACKED

```
struct pseudo\_header PACKED
```

## 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\_destantion

```
uint16_t udp_head::m_port_destantion
```

Port destantion

### 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](#) [IFNAMSIZ]
- 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[IFNAMSIZ]
```

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 <stddef.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\(left, righth\)](#)
- #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\\_destantion\\_udp\\_pack](#) ([udp\\_pack\\_t](#) pack, const char \*const port)  
*Function for setting destantion port in UDP package.*
- ssize\_t [set\\_ip\\_address\\_source\\_udp\\_pack](#) ([udp\\_pack\\_t](#) pack, const char \*const ip)  
*Function for setting source ip address in UDP package.*
- ssize\_t [set\\_ip\\_address\\_destantion\\_udp\\_pack](#) ([udp\\_pack\\_t](#) pack, const char \*const ip)  
*Function for setting destantion ip address 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 for getting size data.*
- `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.*
- `static void * inet_mac (const char *const mac_address)`  
*Function return mac address in big endian.*
- `ssize_t set_mac_address_source_udp_pack (udp_pack_t pack, const char *const mac_address)`  
*Function setting mac address for source.*
- `ssize_t set_mac_address_destantion_udp_pack (udp_pack_t pack, const char *const mac_address)`  
*Function setting mac address for destantion.*
- `char * get_ip_address_source_udp_pack (udp_pack_t pack)`  
*Function for getting ip address source.*
- `char * get_ip_address_destantion_udp_pack (udp_pack_t pack)`  
*Function for getting ip address destantion.*
- `char * get_interface_udp_pack (udp_pack_t pack)`  
*Function for getting interface.*
- `char * get_mac_address_source_udp_pack (udp_pack_t pack)`  
*Function getting mac address for source.*
- `char * get_mac_address_destantion_udp_pack (udp_pack_t pack)`  
*Function setting mac address for destantion.*
- `char * get_port_destantion_udp_pack (udp_pack_t pack)`  
*Function for getting port destantion.*
- `char * get_port_source_udp_pack (udp_pack_t pack)`  
*Function for getting port source.*
- `ssize_t print_udp_pack (udp_pack_t pack)`  
*Function print all info about udp pack.*
- `char * get_data_udp_pack (udp_pack_t pack)`  
*Function for getting data.*
- `void destroy_udp_pack (udp_pack_t pack)`  
*Function free UDP package.*

## Variables

- struct `udp_head` PACKED

### 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,
    right)
```

#### Value:

```
((left) < ((typeof(left))right)) ? (left) : ((typeof(left))right))
```

### 9.3.2.8 NULL\_CHECKSUM

```
#define NULL_CHECKSUM 0x0000
```

## 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_destantion_udp_pack (udp_pack_t pack, const char *const port)`  
*Function for setting destantion port in UDP package.*
- `ssize_t set_ip_address_source_udp_pack (udp_pack_t pack, const char *const ip)`  
*Function for setting source ip address in UDP package.*
- `ssize_t set_ip_address_destantion_udp_pack (udp_pack_t pack, const char *const ip)`  
*Function for setting destantion ip address in 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, 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.*
- `char * get_mac_address_source_udp_pack (udp_pack_t pack)`  
*Function getting mac address for source.*
- `char * get_mac_address_destantion_udp_pack (udp_pack_t pack)`

- `ssize_t set_mac_address_source_udp_pack (udp_pack_t pack, const char *const mac_address)`  
*Function setting mac address for source.*
- `ssize_t set_mac_address_destantion_udp_pack (udp_pack_t pack, const char *const mac_address)`  
*Function setting mac address for destantion.*
- `char * get_interface_udp_pack (udp_pack_t pack)`  
*Function for getting interface.*
- `char * get_port_source_udp_pack (udp_pack_t pack)`  
*Function for getting port source.*
- `char * get_port_destantion_udp_pack (udp_pack_t pack)`  
*Function for getting port destantion.*
- `char * get_ip_address_source_udp_pack (udp_pack_t pack)`  
*Function for getting ip address source.*
- `char * get_ip_address_destantion_udp_pack (udp_pack_t pack)`  
*Function for getting ip address destantion.*
- `char * get_data_udp_pack (udp_pack_t pack)`  
*Function for getting data.*
- `char * get_data_hex_udp_pack (udp_pack_t pack)`  
*Function for getting data hex.*
- `ssize_t print_udp_pack (udp_pack_t pack)`  
*Function print all info about udp pack.*
- `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_destantion_udp_pack(udp_pack_t pack, const char * const port);
00088
00112 ssize_t set_ip_address_source_udp_pack( \
00113     udp_pack_t pack, const char * const ip);
00114
00138 ssize_t set_ip_address_destantion_udp_pack( \
00139     udp_pack_t pack, const char * const ip);
00140
00159 uint16_t get_size_data_udp_pack(udp_pack_t pack);

```

```
00160
00182 ssize_t add_date_udp_pack(udp_pack_t pack, void * data, uint16_t size);
00183
00205 ssize_t set_data_udp_pack(udp_pack_t pack, void * data, \
00206     const uint16_t size);
00207
00228 ssize_t add_byte_udp_pack(udp_pack_t pack, \
00229     const uint8_t byte);
00230
00249 ssize_t set_input_data_udp_pack(udp_pack_t pack);
00250
00270 ssize_t set_file_data_udp_pack( \
00271     udp_pack_t pack, const char * const file_name);
00272
00292 ssize_t set_interface_udp_pack( \
00293     udp_pack_t pack, const char * const interface);
00294
00313 ssize_t send_udp_pack(udp_pack_t pack);
00314
00343 char * get_mac_address_source_udp_pack(udp_pack_t pack);
00344
00373 char * get_mac_address_destantion_udp_pack(udp_pack_t pack);
00374
00398 ssize_t set_mac_address_source_udp_pack(udp_pack_t pack, \
00400     const char * const mac_address);
00400
00424 ssize_t set_mac_address_destantion_udp_pack(udp_pack_t pack, \
00425     const char * const mac_address);
00426
00458 char * get_interface_udp_pack(udp_pack_t pack);
00459
00491 char * get_port_source_udp_pack(udp_pack_t pack);
00492
00524 char * get_port_destantion_udp_pack(udp_pack_t pack);
00525
00557 char * get_ip_address_source_udp_pack(udp_pack_t pack);
00558
00590 char * get_ip_address_destantion_udp_pack(udp_pack_t pack);
00591
00618 uint16_t get_size_data_udp_pack(udp_pack_t pack);
00619
00642 char * get_data_udp_pack(udp_pack_t pack);
00643
00671 char * get_data_hex_udp_pack(udp_pack_t pack);
00672
00695 ssize_t print_udp_pack(udp_pack_t pack);
00696
00714 void destroy_udp_pack(udp_pack_t pack);
00715
```



# 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, 35  
destroy\_udp\_pack  
    work for udp, 15  
  
get\_data\_hex\_udp\_pack  
    work for udp, 15  
get\_data\_udp\_pack  
    work for udp, 16  
get\_interface\_udp\_pack  
    work for udp, 17  
get\_ip\_address\_destantion\_udp\_pack  
    work for udp, 17  
get\_ip\_address\_source\_udp\_pack  
    work for udp, 18  
get\_mac\_address\_destantion\_udp\_pack  
    work for udp, 19  
get\_mac\_address\_source\_udp\_pack  
    work for udp, 19  
get\_pack\_udp\_pack  
    work for udp, 20  
get\_port\_destantion\_udp\_pack  
    work for udp, 20  
get\_port\_source\_udp\_pack  
    work for udp, 21  
get\_size\_data\_udp\_pack  
    work for udp, 22  
  
HEAD\_ETH  
    udp.c, 42  
HEAD\_IP  
    udp.c, 42  
HEAD\_PSEUDO  
    udp.c, 42  
HEAD\_UDP  
    udp.c, 42  
HEAD\_UDP\_IP  
    udp.c, 42  
  
inet\_mac  
    work for udp, 23  
  
inet\_port  
    work for udp, 24  
init\_udp\_pack  
    work for udp, 24  
  
m\_checksum  
    udp\_head, 37  
m\_data  
    udp\_pack, 38  
m\_ethhdr  
    udp\_pack, 38  
m\_head  
    udp\_pack, 38  
m\_interface  
    udp\_pack, 38  
m\_iphdr  
    udp\_pack, 38  
m\_length  
    udp\_head, 37  
m\_port\_destantion  
    udp\_head, 37  
m\_port\_source  
    udp\_head, 37  
main  
    main.c, 39  
main.c, 39  
    main, 39  
MAX\_SIZE\_DATA  
    udp.c, 42  
MIN  
    udp.c, 42  
NULL\_CHECKSUM  
    udp.c, 42  
  
PACKED  
    work for udp, 13, 32  
placeholder  
    pseudo\_header, 35  
print\_udp\_pack  
    work for udp, 24  
protocol  
    pseudo\_header, 35  
pseudo\_header, 35  
    dest\_address, 35  
    placeholder, 35  
    protocol, 35  
    source\_address, 36  
    udp\_length, 36  
  
README.md, 39

**send\_udp\_pack**  
 work for udp, 25  
**set\_data\_udp\_pack**  
 work for udp, 25  
**set\_file\_data\_udp\_pack**  
 work for udp, 26  
**set\_input\_data\_udp\_pack**  
 work for udp, 26  
**set\_interface\_udp\_pack**  
 work for udp, 27  
**set\_ip\_address\_destantion\_udp\_pack**  
 work for udp, 27  
**set\_ip\_address\_source\_udp\_pack**  
 work for udp, 28  
**set\_mac\_address\_destantion\_udp\_pack**  
 work for udp, 29  
**set\_mac\_address\_source\_udp\_pack**  
 work for udp, 29  
**set\_port\_destantion\_udp\_pack**  
 work for udp, 30  
**set\_port\_source\_udp\_pack**  
 work for udp, 30  
**set\_size\_udp\_pack**  
 work for udp, 31  
**source\_address**  
 pseudo\_header, 36  
**sum\_compute**  
 work for udp, 31  
  
**udp**, 1  
**udp.c**  
 HEAD\_ETH, 42  
 HEAD\_IP, 42  
 HEAD\_PSEUDO, 42  
 HEAD\_UDP, 42  
 HEAD\_UDP\_IP, 42  
 MAX\_SIZE\_DATA, 42  
 MIN, 42  
 NULL\_CHECKSUM, 42  
**udp\_head**, 36  
 m\_checksum, 37  
 m\_length, 37  
 m\_port\_destantion, 37  
 m\_port\_source, 37  
**udp\_length**  
 pseudo\_header, 36  
**udp\_lib** Directory Reference, 33  
**udp\_lib/udp.c**, 39  
**udp\_lib/udp.h**, 43, 44  
**udp\_pack**, 37  
 m\_data, 38  
 m\_ethhdr, 38  
 m\_head, 38  
 m\_interface, 38  
 m\_iphdr, 38  
**udp\_pack\_t**  
 work for udp, 13  
  
 work for udp, 11  
  
**add\_byte\_udp\_pack**, 13  
**add\_date\_udp\_pack**, 14  
**calculate\_checksum\_udp\_pack**, 14  
**checksum\_compute**, 15  
**destroy\_udp\_pack**, 15  
**get\_data\_hex\_udp\_pack**, 15  
**get\_data\_udp\_pack**, 16  
**get\_interface\_udp\_pack**, 17  
**get\_ip\_address\_destantion\_udp\_pack**, 17  
**get\_ip\_address\_source\_udp\_pack**, 18  
**get\_mac\_address\_destantion\_udp\_pack**, 19  
**get\_mac\_address\_source\_udp\_pack**, 19  
**get\_pack\_udp\_pack**, 20  
**get\_port\_destantion\_udp\_pack**, 20  
**get\_port\_source\_udp\_pack**, 21  
**get\_size\_data\_udp\_pack**, 22  
**inet\_mac**, 23  
**inet\_port**, 24  
**init\_udp\_pack**, 24  
**PACKED**, 13, 32  
**print\_udp\_pack**, 24  
**send\_udp\_pack**, 25  
**set\_data\_udp\_pack**, 25  
**set\_file\_data\_udp\_pack**, 26  
**set\_input\_data\_udp\_pack**, 26  
**set\_interface\_udp\_pack**, 27  
**set\_ip\_address\_destantion\_udp\_pack**, 27  
**set\_ip\_address\_source\_udp\_pack**, 28  
**set\_mac\_address\_destantion\_udp\_pack**, 29  
**set\_mac\_address\_source\_udp\_pack**, 29  
**set\_port\_destantion\_udp\_pack**, 30  
**set\_port\_source\_udp\_pack**, 30  
**set\_size\_udp\_pack**, 31  
**sum\_compute**, 31  
**udp\_pack\_t**, 13