```
#include "stm32f4xx_conf.h"
#include "ethernet.h"
#include <stdio.h>
#include <string.h>

//source IP
#define S_IP1         192
#define S_IP2         168
#define S_IP3         1
#define S_IP4         100

//dest IP
#define D_IP1         192
#define D_IP2         168
#define D_IP3         1
#define D_IP4         2

struct ip_eth_addr {
  uint8_t addr[4];
};
struct mac_addr {
  uint8_t addr[6];
};

struct ip_eth_addr s_ip = { {S_IP1, S_IP2, S_IP3, S_IP4}};
struct ip_eth_addr d_ip = { {D_IP1, D_IP2, D_IP3, D_IP4}};

struct mac_addr mymac = { {0x1c, 0x9a, 0x65, 0x4a, 0xd9, 0x6c} } ;
struct mac_addr destmac = { {0xff, 0xff, 0xff, 0xff, 0xff, 0xff} } ;
char ipstring [20];

void* first_data_byte;

//define ethernet II header
struct eth_header
{
//ether frame
 uint8_t  ether_dest[6];            //destination mac address
 uint8_t  ether_src[6];                //source mac address
 uint16_t ether_type;                //ether type
//ip frame
 uint8_t vhl;//version / IHL
 uint8_t tos;//type of service
 uint16_t len;//len
 uint16_t ipid;
 uint16_t ipoffset;
 uint8_t  ttl;
 uint8_t  proto;
 uint16_t ipchksum;
 uint16_t srcipaddr[2];  //26
 uint16_t destipaddr[2];
//udp
 uint16_t srcport;
 uint16_t destport;
 uint16_t udplen;
 uint16_t udpchksum;
 };

//struct eth_header frame;
#define frame ((struct eth_header*)&uip_buf[0])
unsigned char frame_counter;
unsigned int test_counter;
char test1[] = "TEST";
unsigned int data_len = 64;            //udp data len
```

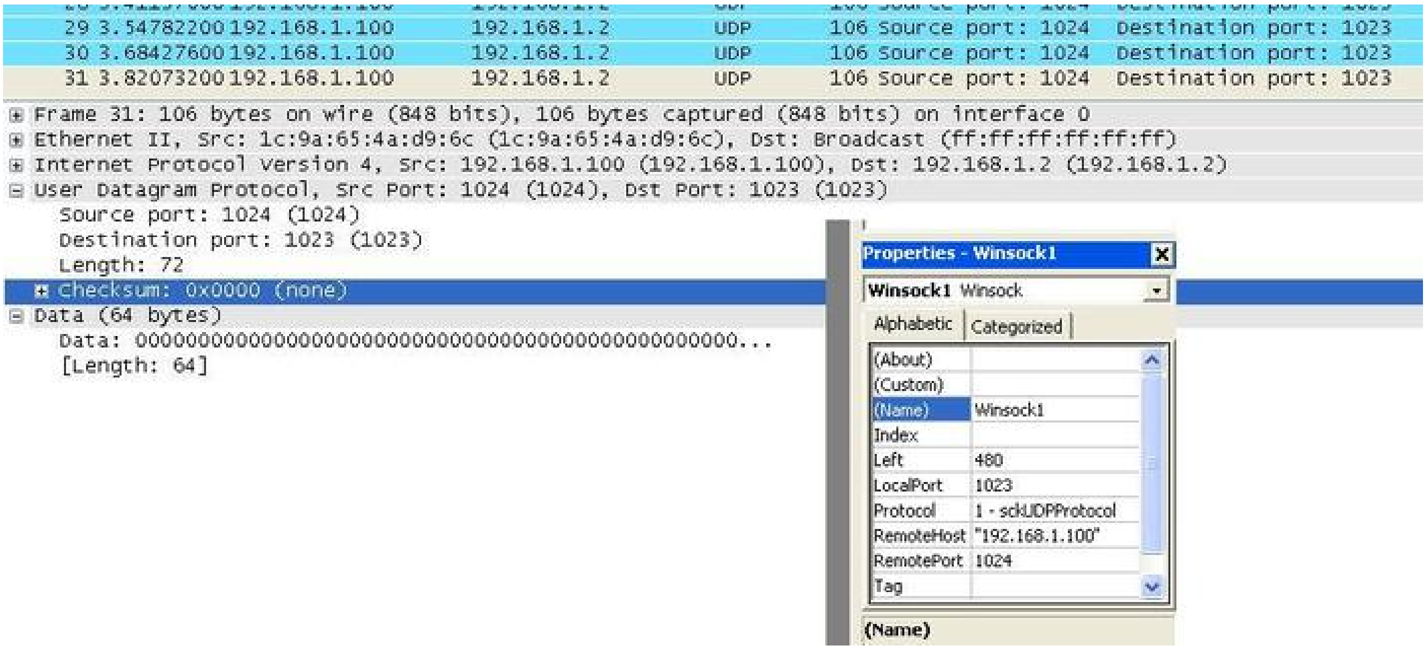
```
void send_UDP_frame(void)
{
 frame_counter++;
 uip_len = sizeof(struct eth_header);    //get struct header len = 42
 memcpy(frame->ether_dest, &destmac, 6);    //0xff = broadcast
 memcpy(frame->ether_src, &mymac, 6);
 frame->ether_type = HTONS(0x0800);        //IP following
 frame->vhl           = 0x45;               //IP4, 20byte header
 frame->tos           = 0x00;               //
 frame->len         = HTONS(28 + data_len);
 frame->ipid        = HTONS(frame_counter);//
 frame->ipoffset    = HTONS(0);
 frame->ttl         = 0x80;             //time to live
 frame->proto         = 0x11;             //UDP (17)
 frame->ipchksum    = 0;                 //reset checksum (will be calculated and inserted later)
 memcpy(frame->srcipaddr, &s_ip,4); //insert source ip
 memcpy(frame->destipaddr,&d_ip,4); //insert dest ip
//calculate IP4 header checksum
 frame->ipchksum=~uip_chksum((uint16_t*)&uip_buf[14],20); //20 byte header checksum

 frame->srcport      = HTONS(1024);    //source port
 frame->destport     = HTONS(1023);    //destination port
 frame->udplen        = HTONS(data_len+8);    //data len + 8 byte udp header
 tapdev_send(uip_buf,uip_len+data_len);
}
```

hello forum,

I am trying to send an empty UDP packet to visual basic 6.0 winsock
as you can see in attached picture wireshark "sees" the UDP packets
however winsock never go into below subroutine (it doesnot get the packets)
what is the missing thing here?

thank you

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Text1 = bytesTotal
Dim strData As String
Winsock1.GetData strData, vbString
End Sub
```



```
    29 3.54782200 192.168.1.100    192.168.1.2     UDP     106 Source port: 1024  Destination port: 1023
    30 3.68427600 192.168.1.100    192.168.1.2     UDP     106 Source port: 1024  Destination port: 1023
    31 3.82073200 192.168.1.100    192.168.1.2     UDP     106 Source port: 1024  Destination port: 1023
⊞ Frame 31: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
⊞ Ethernet II, Src: 1c:9a:65:4a:d9:6c (1c:9a:65:4a:d9:6c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 192.168.1.100 (192.168.1.100), Dst: 192.168.1.2 (192.168.1.2)
⊟ User Datagram Protocol, Src Port: 1024 (1024), Dst Port: 1023 (1023)
    Source port: 1024 (1024)
    Destination port: 1023 (1023)
    Length: 72
  ⊞ Checksum: 0x0000 (none)
⊟ Data (64 bytes)
    Data: 0000000000000000000000000000000000000000000000000...
    [Length: 64]
```

Properties - Winsock1

Winsock1 Winsock

(About)	
(Custom)	
(Name)	Winsock1
Index	
Left	480
LocalPort	1023
Protocol	1 - sckUDPProtocol
RemoteHost	"192.168.1.100"
RemotePort	1024
Tag	

(Name)