

# UDP Client

Coursework Assignment IWT

M G S Dassanayake (12712980)

## Table of Contents

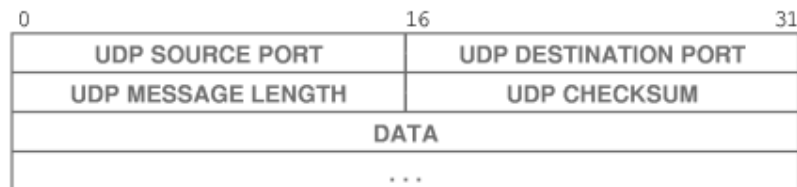
User Datagram Protocol.....	2
PHP Socket Programing.....	3
UDP Client Application .....	4
Bibliography: .....	6
Test Results Transcript 1 .....	7
Test Results Transcript 2 .....	8
Test Results Transcript 3 .....	10
PHP Implementations (Coding).....	11

# User Datagram Protocol

User Datagram protocol is an unreliable and a connectionless protocol. It does not guarantee reliability or the correct sequencing of data. Datagrams may go missing without notice, or arrive in a different order from the one in which they were sent

## UDP Datagram Format

UDP provides a way for applications to send encapsulated IP datagrams without having to establish a connection. UDP transmits *segments* consisting of an 8-byte header followed by the payload. The format is shown below



## UDP Packet Structure

The UDP header consists of 4 fields, each of which is 2 bytes (16 bits). The use of two of those is optional in IPv4 (pink background in table). In IPv6 only the source port is optional (see below).

### Source port number

This field identifies the sender's port when meaningful and should be assumed to be the port to reply to if needed. If not used, then it should be zero. If the source host is the client, the port number is likely to be an ephemeral port number. If the source host is the server, the port number is likely to be a well-known port number.

### Destination port number

This field identifies the receiver's port and is required. Similar to source port number, if the client is the destination host then the port number will likely be an ephemeral port number and if the destination host is the server then the port number will likely be a well-known port number.

### Length

A field that specifies the length in bytes of the entire datagram: header and data. The minimum length is 8 bytes since that's the length of the header. The field size sets a theoretical limit of 65,535 bytes (8 byte header + 65,527 bytes of data) for a UDP datagram. The practical limit for the data length which is imposed by the underlying IPv4 protocol is 65,507 bytes (65,535 – 8 byte UDP header – 20 byte IP header).

### Checksum

The checksum field is used for error-checking of the header *and* data. If no checksum is generated by the transmitter, the field uses the value all-zeros. This field is not optional for IPv6.

# PHP Socket Programing

PHP comes with a comprehensive socket programming API which has all the functions you need to create socket based client – server application.

Functions I have used in my program

<code>socket_create()</code>	Create a socket (endpoint for communication)
<code>socket_connect()</code>	Initiates a connection on a socket
<code>socket_sendto()</code>	Sends a message to a socket, whether it is connected or not
<code>socket_recvfrom()</code>	Receives data from a socket whether or not it is connection-oriented
<code>socket_close(\$sock)</code>	Closes a socket resource

# UDP Client Application

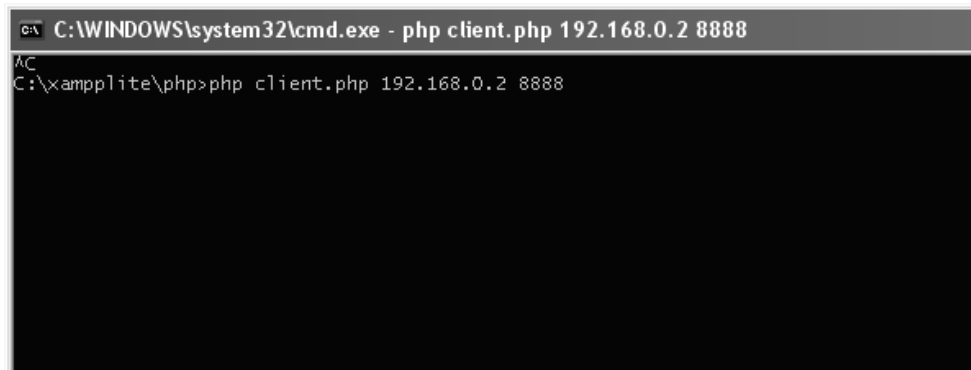
To run the UDP application the user need to execute the client.php file with two arguments in the command prompt.

Argument 1: Server IP

Argument 2: Port Number

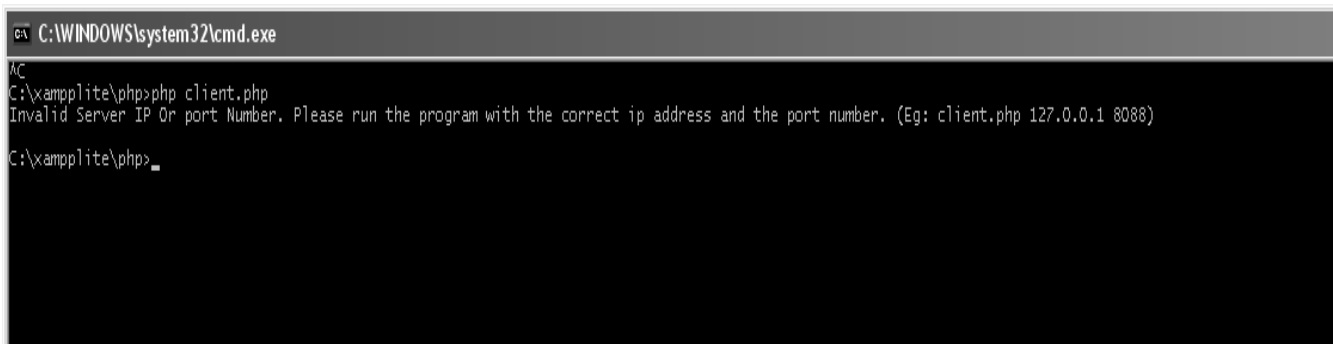
\$php>php client.php <server\_ip> <server\_port>

(Eg: \$php>php client.php 192.168.0.2 8888)



```
C:\WINDOWS\system32\cmd.exe - php client.php 192.168.0.2 8888
C:\xampplite\php>php client.php 192.168.0.2 8888
```

If the user run the php file without any arguments the program will display an error message.



```
C:\WINDOWS\system32\cmd.exe
C:\xampplite\php>php client.php
Invalid Server IP Or port Number. Please run the program with the correct ip address and the port number. (Eg: client.php 127.0.0.1 8088)
C:\xampplite\php>
```

If the IP address is not valid it will display the following error message

```
C:\WINDOWS\system32\cmd.exe

C:\xampplite\php>php client.php 259.168.0.2 8888

Warning: socket_connect(): Host lookup failed [0]: No such host is known.
in C:\xampplite\php\client.php on line 30

C:\xampplite\php>
```

On my system I have installed xampplite which is a software bundle which includes PHP 5.3.1, Apache, MySQL and many other services.

For the development and for testing purposes I have used the PHP interpreter came with the xampplite budle.

Once you run the application it will wait for a keyboard input. (As shown in the above figure)

If the server is running on the given IP and the Port number it will display the input text with the server IP and the port number from the datagram received back from the server. (Test results are provided in the Appendix)

If the server is not running or if the program is not able to connect to the server it will display the following warning message

```
C:\WINDOWS\system32\cmd.exe - php client.php 192.168.0.2 8888

AC
C:\xampplite\php>php client.php 192.168.0.2 8888
Hello

Warning: socket_recvfrom(): unable to recvfrom [0]: The operation completed successfully.
in C:\xampplite\php\client.php on line 42
192.168.0.2 8888
```

To exit the application type the word 'quit' instead of the message.

## **Bibliography:**

<http://php.net/manual/en/function.socket-recvfrom.php>

<http://penguin.dcs.bbk.ac.uk/academic/networks/transport-layer/udp/index.php>

[http://en.wikipedia.org/wiki/User\\_Datagram\\_Protocol](http://en.wikipedia.org/wiki/User_Datagram_Protocol)

<http://www.php.net/manual/en/ref.sockets.php>

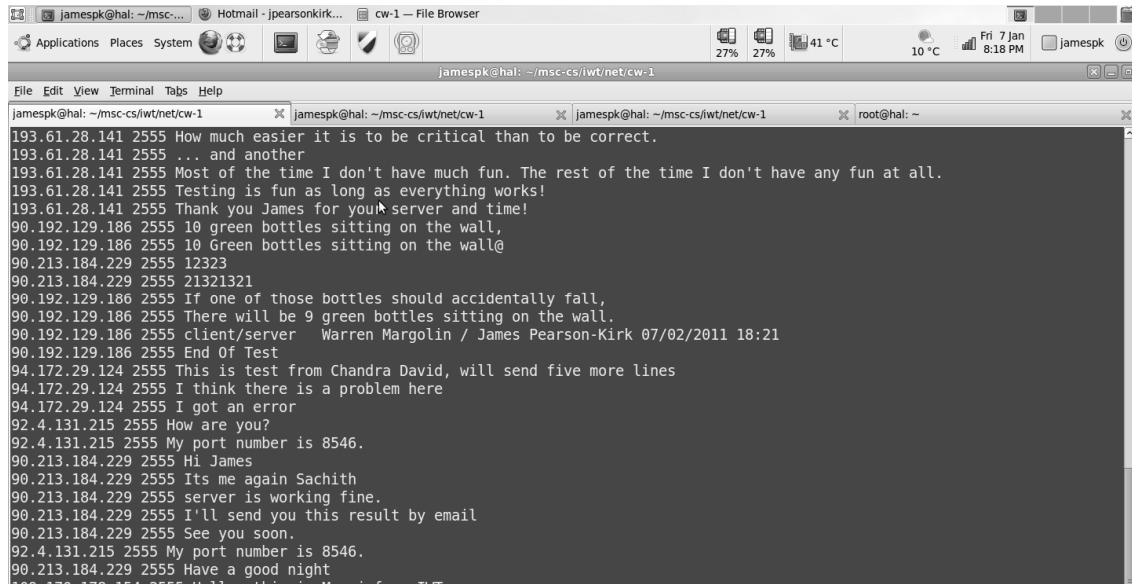
<http://www.apachefriends.org/en/xampp-windows.html>

## Test Results Transcript 1

Name: James Pearson-Kirk (07<sup>th</sup> January 2010)

Server IPAddress: 86.14.117.160

Server Port Number: 2555

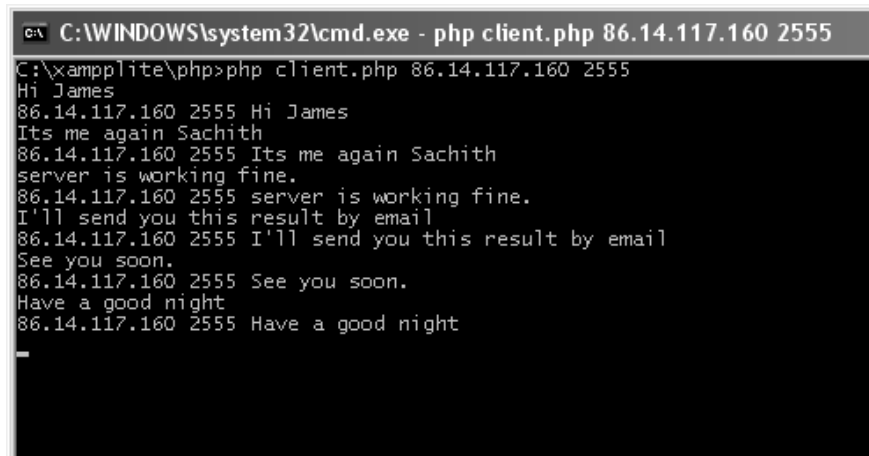


```
jamespk@hal: ~/msc-cs/iwt/net/cw-1
193.61.28.141 2555 How much easier it is to be critical than to be correct.
193.61.28.141 2555 ... and another
193.61.28.141 2555 Most of the time I don't have much fun. The rest of the time I don't have any fun at all.
193.61.28.141 2555 Testing is fun as long as everything works!
193.61.28.141 2555 Thank you James for your server and time!
90.192.129.186 2555 10 green bottles sitting on the wall,
90.192.129.186 2555 10 Green bottles sitting on the wall@
90.213.184.229 2555 12323
90.213.184.229 2555 21321321
90.192.129.186 2555 If one of those bottles should accidentally fall,
90.192.129.186 2555 There will be 9 green bottles sitting on the wall.
90.192.129.186 2555 client/server Warren Margolin / James Pearson-Kirk 07/02/2011 18:21
90.192.129.186 2555 End Of Test
94.172.29.124 2555 This is test from Chandra David, will send five more lines
94.172.29.124 2555 I think there is a problem here
94.172.29.124 2555 I got an error
92.4.131.215 2555 How are you?
92.4.131.215 2555 My port number is 8546.
90.213.184.229 2555 Hi James
90.213.184.229 2555 Its me again Sachith
90.213.184.229 2555 server is working fine.
90.213.184.229 2555 I'll send you this result by email
90.213.184.229 2555 See you soon.
92.4.131.215 2555 My port number is 8546.
90.213.184.229 2555 Have a good night
```

Figure 1: UDP Server

Client IP Address: 90.213.184.229

Client Port : 2555 (This was number received from the server as the port)



```
C:\WINDOWS\system32\cmd.exe - php client.php 86.14.117.160 2555
C:\xampp\lite\php>php client.php 86.14.117.160 2555
Hi James
86.14.117.160 2555 Hi James
Its me again Sachith
86.14.117.160 2555 Its me again Sachith
server is working fine.
86.14.117.160 2555 server is working fine.
I'll send you this result by email
86.14.117.160 2555 I'll send you this result by email
See you soon.
86.14.117.160 2555 See you soon.
Have a good night
86.14.117.160 2555 Have a good night
```

UDP Client



## Test Results Transcript 2

Name: Michael Sauter (08<sup>th</sup> January 2010)

Server IPAddress: 192.168.0.2

Server Port Number: 9000

```
C:\WINDOWS\system32\cmd.exe - php client.php 192.168.0.2 9000
C:\xampplite\php>php client.php 192.168.0.2 9000
Hello Michael
192.168.0.2 9000 Hello Michael
This is Sachith
192.168.0.2 9000 This is Sachith
Testing the UDP Server
192.168.0.2 9000 Testing the UDP Server
I'm getting the messages back
192.168.0.2 9000 I'm getting the messages back
Looks like its working
192.168.0.2 9000 Looks like its working
See you on Monday
192.168.0.2 9000 See you on Monday
Have a good day
192.168.0.2 9000 Have a good day
```

UDP Client

Client IP Address: 192.168.0.2

Client Port: 3345

```
C:\WINDOWS\system32\cmd.exe - java UDPServer 9000
C:\Program Files\Java\jre6\bin>java UDPServer 9000
/192.168.0.2:3345 Hello Michael

/192.168.0.2:3345 This is Sachith

/192.168.0.2:3345 Testing the UDP Server

/192.168.0.2:3345 I'm getting the messages back

/192.168.0.2:3345 Looks like its working

/192.168.0.2:3345 See you on Monday

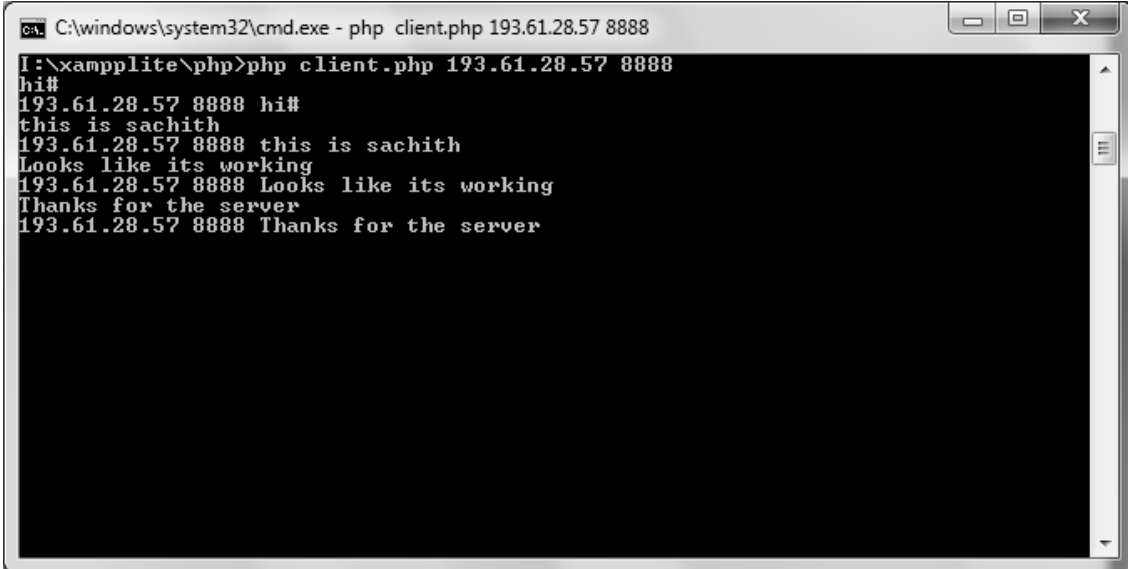
/192.168.0.2:3345 Have a good day
```

UDP Server

The same server was also tested on a lab computer and the results as follows.

Server IPAddress: 193.61.28.57 (10<sup>th</sup> January 2010)

Server Port Number: 8888



A screenshot of a Windows command prompt window. The title bar reads "C:\windows\system32\cmd.exe - php client.php 193.61.28.57 8888". The command prompt shows the following text:

```
I:\xampplite\php>php client.php 193.61.28.57 8888
hi#
193.61.28.57 8888 hi#
this is sachith
193.61.28.57 8888 this is sachith
Looks like its working
193.61.28.57 8888 Looks like its working
Thanks for the server
193.61.28.57 8888 Thanks for the server
```

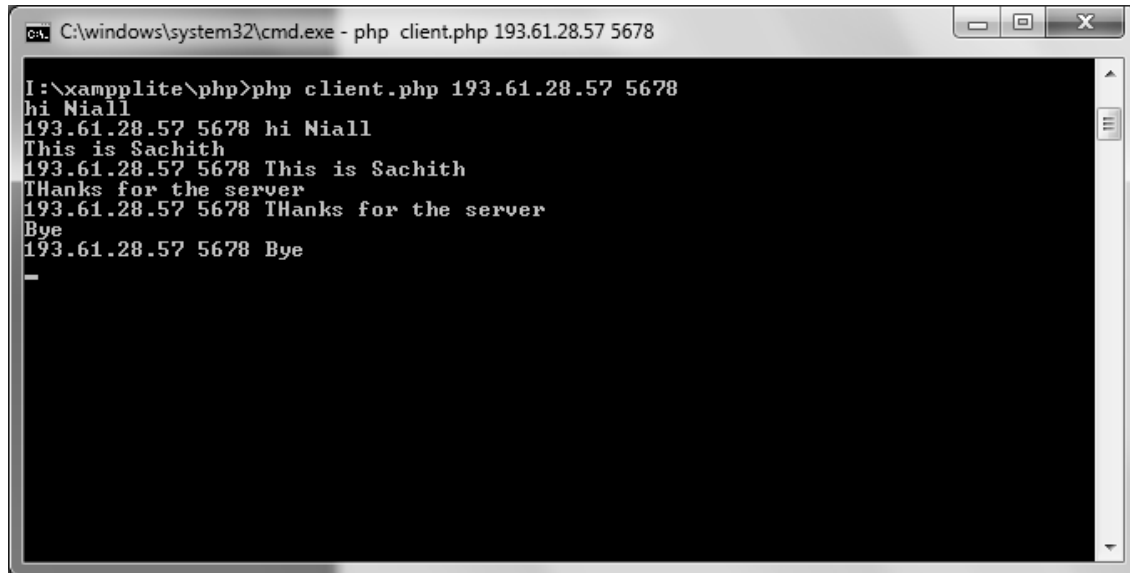
UDP Client

## Test Results Transcript 3

Name: Niall Gallagher (12<sup>th</sup> January 2010)

Server IP Address: 193.61.28.57

Server Port Number: 5678



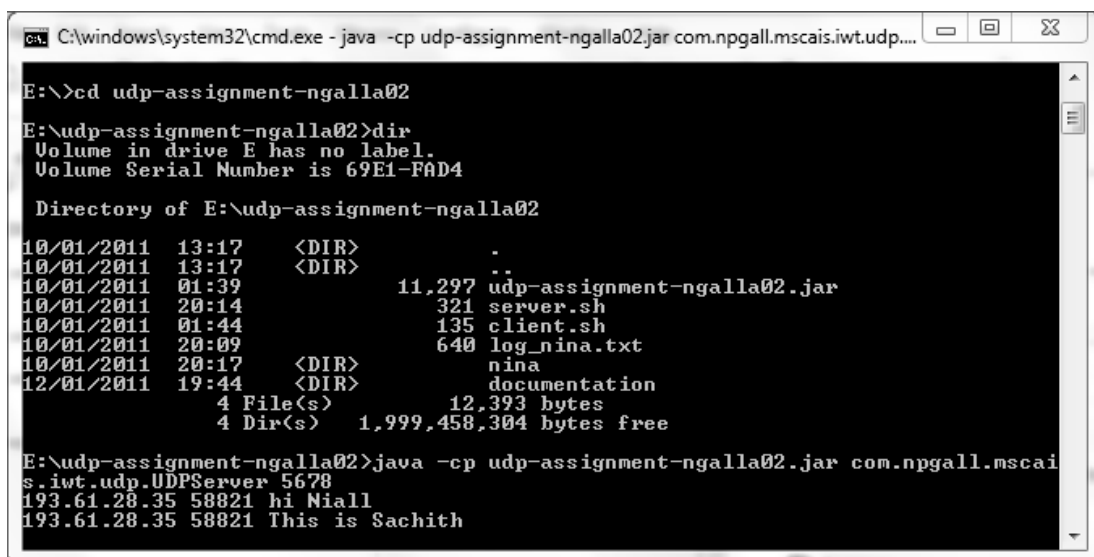
A screenshot of a Windows command prompt window. The title bar reads "C:\windows\system32\cmd.exe - php client.php 193.61.28.57 5678". The command prompt shows the following text:

```
I:\xampplite\php>php client.php 193.61.28.57 5678
hi Niall
193.61.28.57 5678 hi Niall
This is Sachith
193.61.28.57 5678 This is Sachith
THanks for the server
193.61.28.57 5678 THanks for the server
Bye
193.61.28.57 5678 Bye
```

UDP Client

Client IP Address: 193.61.28.35

Port Number: 58821



A screenshot of a Windows command prompt window. The title bar reads "C:\windows\system32\cmd.exe - java -cp udp-assignment-ngalla02.jar com.npgall.mscais.iwt.udp....". The command prompt shows the following text:

```
E:\>cd udp-assignment-ngalla02
E:\udp-assignment-ngalla02>dir
Volume in drive E has no label.
Volume Serial Number is 69E1-FAD4

Directory of E:\udp-assignment-ngalla02

10/01/2011 13:17 <DIR> .
10/01/2011 13:17 <DIR> ..
10/01/2011 01:39 11,297 udp-assignment-ngalla02.jar
10/01/2011 20:14 321 server.sh
10/01/2011 01:44 135 client.sh
10/01/2011 20:09 640 log_nina.txt
10/01/2011 20:17 <DIR> nina
12/01/2011 19:44 <DIR> documentation
4 File(s) 12,393 bytes
4 Dir(s) 1,999,458,304 bytes free

E:\udp-assignment-ngalla02>java -cp udp-assignment-ngalla02.jar com.npgall.mscai
s.iwt.udp.UDPServer 5678
193.61.28.35 58821 hi Niall
193.61.28.35 58821 This is Sachith
```

UDP Server

## ***PHP Implementations (Coding)***

```
<?php
```

```
/* Coursework Assignment IWT
```

```
    UDP Client
```

```
    M G S Dassanayake (12712980)
```

```
    References: Example #1 http://php.net/manual/en/function.socket-recvfrom.php
```

```
*/
```

```
$srvIP = $argv[1]; // Remote Server IP address (Argument 1)
```

```
$srvPort = $argv[2]; // Remote Server Port (Argument 2)
```

```
if ($argv[1] == "" || $argv[2] == "")
```

```
{
```

```
    echo "Invalid Server IP Or port Number. Please run the program with the correct ip address and the port  
    number. (Eg: client.php 127.0.0.1 8088)";
```

```
}
```

```
else
```

```
{
```

```
    //Creates the UDP Socket
```

```
    $sock = socket_create(AF_INET, SOCK_DGRAM, SOL_UDP);
```

```
    if(!$sock)
```

```
    {
```

```
        echo 'Socket_create failed: '.socket_strerror(socket_last_error())."\n";
```

```
    }
```

```
    else
```

```
    {
```

```
        // Socket Connect
```

```
        $result = socket_connect($sock, $srvIP, $srvPort);
```

```
        if($result)
```

```
        {
```

```

do{

    // Keyboard Input

    $msg = trim(fgets(STDIN));

    // Check if the user wants to close the application

    if ($msg!= "quit")

    {

        // Length of the input

        $len = strlen($msg);

        // Sends the input message to the server

        socket_sendto($sock, $msg, $len, 0, $srvIP, $srvPort);

        // Receives the message from the server

        socket_recvfrom($sock, $buf, 2048, 0, $srvIP, $srvPort);

        //Shows the message received from the server including the ip address and
the port

        echo $srvIP . " " . $srvPort. " " . trim($buf) . PHP_EOL;

        //Clears the buffer

        $buf = "";

    }

}while($msg!= "quit"); // Exit the application when the user type 'quit'

}

}

//Close the socket

socket_close($sock);

}

?>

```