```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <unistd.h>
#include <arpa/inet.h>


int main(int argc, char *argv[])
{
  int sock;
  struct sockaddr_in sa;
  int bytes_sent;
  char buffer[200];
  char addr[200];
   uint16_t port;
if(argc<4) {fprintf(stderr,"usage: client addr port message\n"); return 0;};
sscanf(argv[2], "%d",&port);
sscanf(argv[1], "%s",addr);
sscanf(argv[3],"%s",buffer);
//const char* buffer = "Привет!";

//printf("%s\n",addr);
//printf("%s\n",buffer);
//printf("%d\n",port);

  //create an internet, datagram, socket using UDP
  sock = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
  if (-1 == sock) /* if socket failed to initialize, exit */
    {
      printf("Error Creating Socket");
      exit(EXIT_FAILURE);
    }

  //Zero out socket address
  memset(&sa, 0, sizeof sa);

  //The address is ipv4
  sa.sin_family = AF_INET;

   //ip_v4 adresses is a uint32_t, convert a string representation of the octets to the
appropriate value
  sa.sin_addr.s_addr = inet_addr(addr);

  //sockets are unsigned shorts, htons(x) ensures x is in network byte order, set the
port to 7654
  sa.sin_port = htons(port);

  //sendto(int socket, char data, int dataLength, flags, destinationAddress, int
destinationStructureLength)
  bytes_sent = sendto(sock, buffer, strlen(buffer), 0,(struct sockaddr*)&sa, sizeof sa);
  if (bytes_sent < 0) {
    printf("Error sending packet: %s\n", strerror(errno));
    exit(EXIT_FAILURE);
  }

  close(sock); /* close the socket */
  return 0;
}
```