

The logo for RT-Thread, featuring a stylized blue wave above the text "RT-Thread" in a bold, blue, sans-serif font. The logo is centered within a white, 3D-style circular shape that has a slight shadow and is surrounded by several smaller orange and white circles of varying sizes, creating a dynamic, modern feel.

RT-Thread

# 网络编程入门篇

使用 socket 实现 UDP 客户端

# 目录

- 基础知识
- 具体示例
- 示例代码讲解



# 基础知识

# 基础知识

- 和TCP协议一样，UDP 协议也是用于客户端-服务器模式的一种传输协议，如今的很多通信软件都是利用这个协议实现的，如腾讯 QQ 发送消息,视频聊天用的就是 UDP 协议。
- 今天我们要讲的就是如何利用 socket 编程实现基于 UDP 协议通信的客户端与服务器进行通信。
- 与开发 TCP 客户端一样，我们先将 socket 编程的流程列出来，然后给出具体的示例，最后讲解一下示例代码。

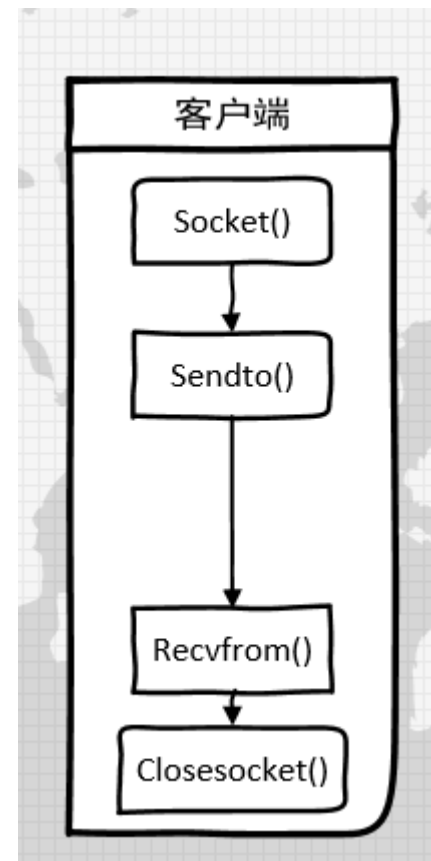
# 基础知识

UDP 与 TCP 的不同之处是，他的通信不需要建立连接的过程。

UDP 客户端的 socket 编程流程

- 创建 socket
- 通信
- 关闭 socket

如右图所示：





# 具体示例

# 工程配置

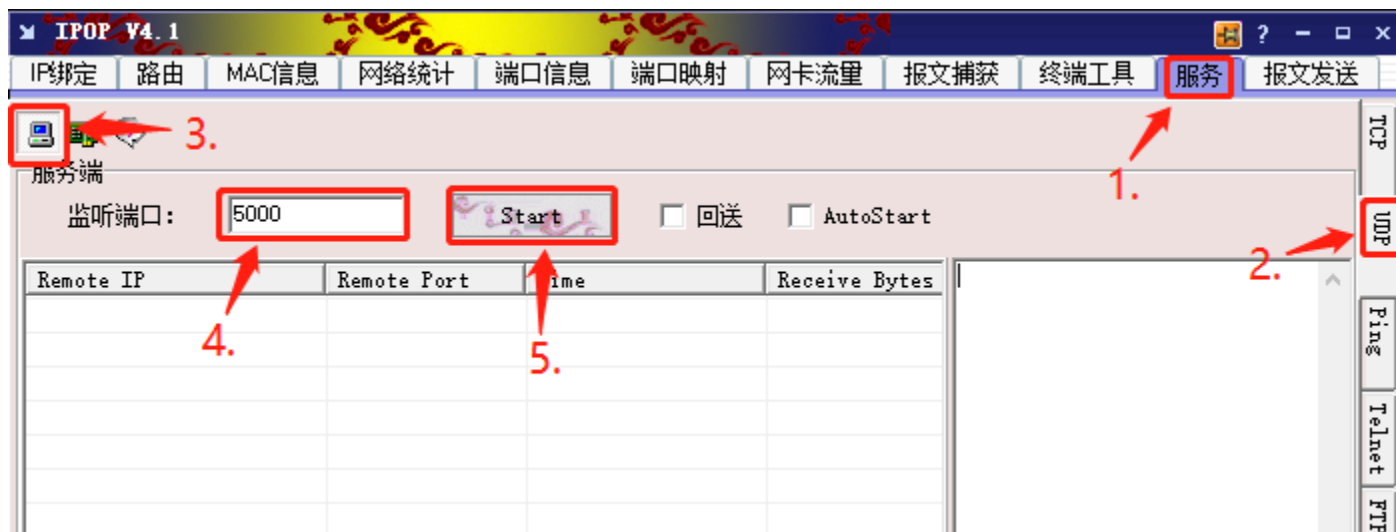
- RT-Thread samples 软件包中已有一份该示例代码 `udpclient.c`，可以通过 `env` 配置将示例代码加入到项目中。
- 按照下面的路径获取 `udp client` 的示例代码：

```
RT-Thread online packages --->
  miscellaneous packages --->
    samples: RT-Thread kernel and components samples --->
      network sample options --->
        [*] [network] udp client
```

- 保存并更新软件包 `pkgs --update`
- 编译工程 `scons`
- 然后运行 `qemu`

# 开启TCP服务器

- 在运行示例代码之前需要先在电脑上开启一个 UDP 服务器，这里以网络调试助手 IPOP 为例。





# 查看本机ip地址

- 在windows系统中打开命令提示符，输入ipconfig即可查看本机ip

```
管理员: 命令提示符
C:\WINDOWS\system32>
C:\WINDOWS\system32>ipconfig

Windows IP 配置

以太网适配器 以太网 2:

    连接特定的 DNS 后缀 . . . . . :
    IPv6 地址 . . . . . : 2001:470:d:aa9:a945:38e7:1ee2:1119
    临时 IPv6 地址. . . . . : 2001:470:d:aa9:14fb:d705:f25a:fb26
    本地链接 IPv6 地址. . . . . : fe80::a945:38e7:1ee2:1119%7
    IPv4 地址 . . . . . : 192.168.12.44
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : fe80::ce2d:e0ff:fe05:3b5a%7
                        192.168.10.1

以太网适配器 tap:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址. . . . . : fe80::d462:3cb4:193:1998%3
    IPv4 地址 . . . . . : 192.168.137.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . :
```

# 运行示例代码

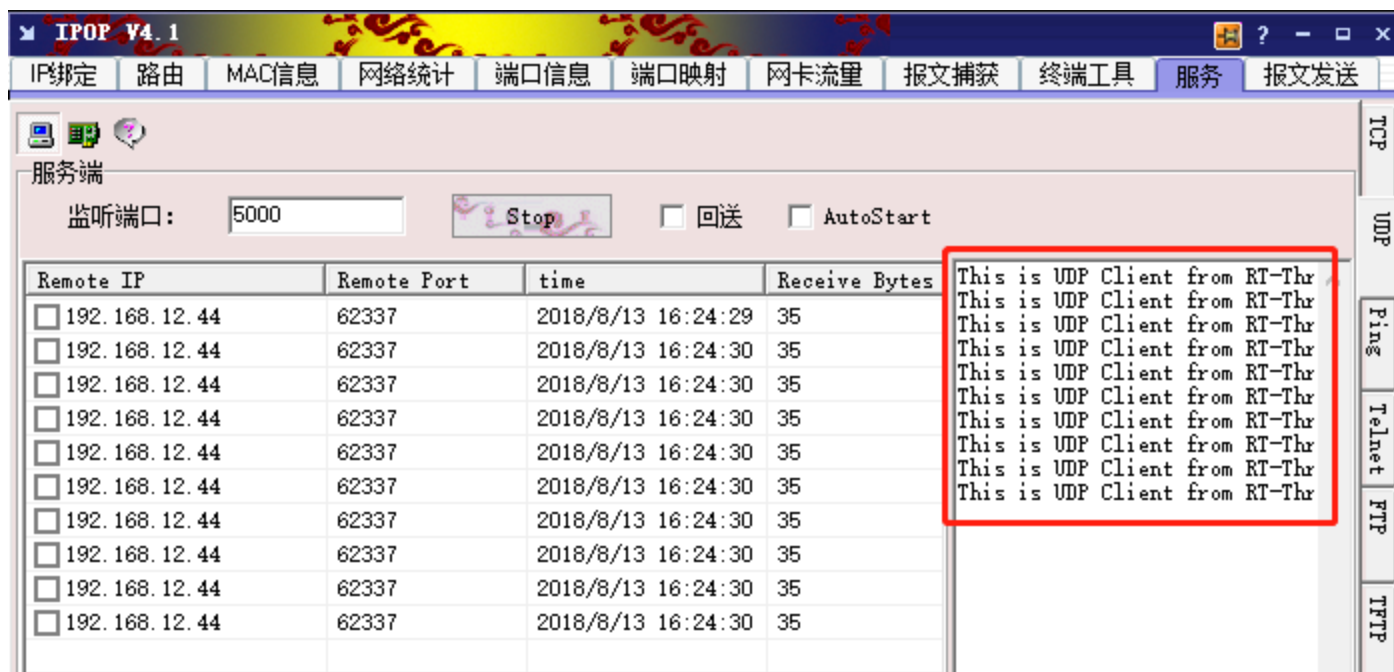
- 在qemu运行起来后，在 msh 命令行下输入下面的命令即可让示例代码运行。

```
msh> udpclient 192.168.12.44 5000
```

- 这个示例代码的功能是向输入的 IP 地址发送 10 条消息，发送完毕即退出，具体的参数如下。
- **udpclient** 有两个固定参数和一个可选参数 **URL PORT [COUNT = 10]**
- 其中：
  - **URL** 是目标服务器的网址或 IP 地址，这个对应刚才搭建的服务器中的本机地址
  - **PORT** 是目标服务器的端口号，这个对应刚才输出的端口号
  - **[COUNT = 10]** 是向服务器发送数据的条数 默认是10

# 预期结果

从电脑搭建的 UDP 服务器上能接收到客户端发来的10条信息



The screenshot shows the IPOP V4.1 interface with the 'Service' (服务) tab selected. The 'Listen Port' (监听端口) is set to 5000. The log table displays 10 entries, all from Remote IP 192.168.12.44 and Remote Port 62337, with a message length of 35 bytes. The messages are: 'This is UDP Client from RT-Thr'. A red box highlights the message content for the last 10 rows.

Remote IP	Remote Port	time	Receive Bytes	Message
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:29	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr
<input type="checkbox"/> 192.168.12.44	62337	2018/8/13 16:24:30	35	This is UDP Client from RT-Thr

# 注意事项

- 电脑需要关闭防火墙



# 示例代码讲解

# 示例代码讲解

```
/*
 * 程序清单： udp 客户端
 *
 * 这是一个 udp 客户端的例程
 * 导出 udpclient 命令到控制终端
 * 命令调用格式： udpclient URL PORT [COUNT = 10]
 * URL： 服务器地址 PORT： 端口号 COUNT： 可选参数 默认为 10
 * 程序功能： 发送 COUNT 条数据到服务远端
 */
#include <rtthread.h>
#include <sys/socket.h> /* 使用BSD socket， 需要包含sockets.h头文件 */
#include <netdb.h>
#include <string.h>
const char send_data[] = "This is UDP Client from RT-Thread.\n"; /* 发送用到的数据 */
void udpclient(int argc, char **argv)
{
    int sock, port, count;
    struct hostent *host;
    struct sockaddr_in server_addr;
    const char *url;
    if (argc < 3)
    {
        rt_kprintf("Usage: udpclient URL PORT [COUNT = 10]\n");
        rt_kprintf("Like: tcpclient 192.168.12.44 5000\n");
        return ;
    }
}
```

# 示例代码讲解

```
url = argv[1];
port = strtoul(argv[2], 0, 10);

if (argc > 3)
    count = strtoul(argv[3], 0, 10);
else
    count = 10;

/* 通过函数入口参数url获得host地址 (如果是域名, 会做域名解析) */
host = (struct hostent *) gethostbyname(url);

/* 创建一个socket, 类型是SOCK_DGRAM, UDP类型 */
if ((sock = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
{
    rt_kprintf("Socket error\n");
    return;
}

/* 初始化预连接的服务端地址 */
server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(port);
server_addr.sin_addr = *((struct in_addr *) host->h_addr);
rt_memset(&(server_addr.sin_zero), 0, sizeof(server_addr.sin_zero));
```

# 示例代码讲解

```
/* 总计发送count次数据 */
while (count)
{
    /* 发送数据到服务远端 */
    sendto(sock, send_data, strlen(send_data), 0,
           (struct sockaddr *)&server_addr, sizeof(struct sockaddr));

    /* 线程休眠一段时间 */
    rt_thread_delay(50);

    /* 计数值减一 */
    count--;
}

/* 关闭这个socket */
closesocket(sock);
}
MSH_CMD_EXPORT(udpclient, a udp client sample);
```