

TCP/IP 网络组件 Lwip 之 TCP Client

RealTouch 评估板 RT-Thread 入门文档

版本号：1.0.0

日期：2012/8/29

修订记录

日期	作者	修订历史
2012/8/29	bloom5	创建文档

实验目的

- ☐ 快速了解 Lwip 组件。
- ☐ 了解 TCP 传输协议的应用，熟悉 TCP Server 和 TCP Client 之间的通信机制。

硬件说明

本实验使用 RT-Thread 官方的 Realtouch 开发板作为实验平台。涉及到的硬件主要为

- ☐ RJ45 接口，作为网络连接的需要，我们需要用网线将 Realtouch 和目标机连接起来，具体请参见《Realtouch 开发板使用手册》
- ☐ 串口 3，作为 rt_kprintf 输出，需要连接 JTAG 扩展板

实验原理及程序结构

实验设计

TCP 全称为 Transmission Control Protocol，中文名为传输控制协议。TCP 是一种面向连接（连接导向）的、可靠的、基于字节流的运输层（Transport layer）通信协议，由 IETF 的 RFC 793 说明（specified）。在简化的计算机网络 OSI 模型中，它完成第四层传输层所指定的功能。

本实验中 RealTouch 作为 TCP Client，PC 机作为 TCP Server，两机进行网络通信。

源程序说明

系统依赖

在 rtconfig.h 中需要开启

- ☐ #define RT_USING_HEAP

此项可选，开启此项可以创建动态线程和动态信号量，如果使用静态线程和静态信号量，则此项不是必要的

- ☐ #define RT_USING_LWIP

此项必须，本实验使用 LWIP 组件，因此需要开启此项

- ☐ #define RT_USING_CONSOLE

此项必须，在开始过程中仍需通过串口进行显示相关的工作

主程序说明

首先，在初始化线程中完成了网络驱动的初始化，lwip 初始化，然后开启了 tcpsrv。

```
void rt_init_thread_entry(void* parameter)
{
#ifdef RT_USING_LWIP
    /* initialize eth interface */
    rt_hw_stm32_eth_init();
#endif

#ifdef RT_USING_COMPONENTS_INIT
    /* initialization RT-Thread Components */
    rt_components_init();
#endif

    rt_platform_init();
    /* do some thing here. */
    tcpclient("192.168.1.11", 4663);
}

int rt_application_init()
{
    rt_thread_t init_thread;

    init_thread = rt_thread_create("init",
                                    rt_init_thread_entry, RT_NULL,
                                    2048, 8, 20);

    if (init_thread != RT_NULL)
        rt_thread_startup(init_thread);

    return 0;
}
```

tcpcli 所有的操作均在 tcpclient.c 中的 tcpclient() 函数中完成，源码配有详尽中文注释，可以帮助你进一步了解其工作机制。

```
#include <rtthread.h>
#include <lwip/netdb.h> /* 为了解析主机名，需要包含 netdb.h 头文件 */
#include <lwip/sockets.h> /* 使用 BSD socket，需要包含 sockets.h 头文件 */

#define BUFSZ 1024
```

```

static const char send_data[] = "This is TCP Client from
RT-Thread."; /* 发送用到的数据 */
void tcpclient(const char *url, int port)
{
    char *recv_data;
    struct hostent *host;
    int sock, bytes_received;
    struct sockaddr_in server_addr;

    /* 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 = gethostbyname(url);

    /* 分配用于存放接收数据的缓冲 */
    recv_data = rt_malloc(BUFSZ);
    if (recv_data == RT_NULL)
    {
        rt_kprintf("No memory\n");
        return;
    }

    /* 创建一个 socket，类型是 SOCKET_STREAM，TCP 类型 */
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    {
        /* 创建 socket 失败 */
        rt_kprintf("Socket error\n");

        /* 释放接收缓冲 */
        rt_free(recv_data);
        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));

    /* 连接到服务端 */
    if (connect(sock, (struct sockaddr *)&server_addr,
sizeof(struct sockaddr)) == -1)
    {
        /* 连接失败 */

```

```

rt_kprintf("Connect error\n");

/*释放接收缓冲 */
rt_free(recv_data);
return;
}

while (1)
{
    /* 从 sock 连接中接收最大 BUFSZ - 1 字节数据 */
    bytes_received = recv(sock, recv_data, BUFSZ - 1, 0);
    if (bytes_received <= 0)
    {
        /* 接收失败，关闭这个连接 */
        lwip_close(sock);

        /* 释放接收缓冲 */
        rt_free(recv_data);
        break;
    }

    /* 有接收到数据，把末端清零 */
    recv_data[bytes_received] = '\0';

    if (strcmp(recv_data , "q") == 0 || strcmp(recv_data , "Q")
== 0)
    {
        /* 如果是首字母是 q 或 Q，关闭这个连接 */
        lwip_close(sock);

        /* 释放接收缓冲 */
        rt_free(recv_data);
        break;
    }
    else
    {
        /* 在控制终端显示收到的数据 */
        rt_kprintf("\nRecieved data = %s " , recv_data);
    }

    /* 发送数据到 sock 连接 */
    send(sock, send_data, strlen(send_data), 0);
}

```

```
    return;
}

#ifdef RT_USING_FINSH
#include <finsh.h>
/* 输出 tcpclient 函数到 finsh shell 中 */
FINSH_FUNCTION_EXPORT(tcpclient, startup tcp client);
#endif
```

值得注意的是 tcpclient.c 16

```
host = gethostbyname(url);
```

url 为传入的参数,可以是域名也可以直接是 IP 地址,所以 tcpclient() 函数可以设置为如下形式:

```
tcpclient("192.168.1.52",4663);
tcpclient("baidu.com",80);
```

编译调试及观察输出信息

编译请参见《RT-Thread 配置开发环境指南》完成编译烧录,参考《Realtouch 开发板使用手册》完成硬件连接,连接好串口线,连上网线。首先在运行板上程序之前要做的是打开 PC 上的 TCP Server 客户端:



接着运行板上程序后可以看到串口有如下的信息：

```
\ | /
- RT -      Thread Operating System
/ | \      1.1.0 build Aug 29 2012
2006 - 2012 Copyright by rt-thread team
TCP/IP initialized!
finsh>>
```

但是！！！现在你不会看到任何现象。因为！！请看：

```
while (1)
{
    /* 从 sock 连接中接收最大 BUFSZ - 1 字节数据 */
    bytes_received = recv(sock, recv_data, BUFSZ - 1, 0);
    if (bytes_received <= 0)
    {
        /* 接收失败，关闭这个连接 */
        lwip_close(sock);
    }
}
```

```

        /* 释放接收缓冲 */
        rt_free(recv_data);
        break;
    }

    /* 有接收到数据，把末端清零 */
    recv_data[bytes_received] = '\0';

    if (strcmp(recv_data, "q") == 0 || strcmp(recv_data, "Q")
== 0)
    {
        /* 如果是首字母是 q 或 Q，关闭这个连接 */
        lwip_close(sock);

        /* 释放接收缓冲 */
        rt_free(recv_data);
        break;
    }
    else
    {
        /* 在控制终端显示收到的数据 */
        rt_kprintf("\nRecieved data = %s " , recv_data);
    }

    /* 发送数据到 sock 连接 */
    send(sock, send_data, strlen(send_data), 0);
}

```

只有你从 Server 端发送数据以后，Client 才回返回相应数据：



同时串口上我们可以看到：

Recieved data = a

结果分析

通过以上内容的实际操作，实现了 TCP server 与 client 的连接，并且进行了数据通信。