

RT1060 UDP 组播代码测试

一， 文档说明

本篇文章源自客户需要在 RT4 位芯片 SDK LWIP UDP 代码中添加 UDP 组播功能，遇到问题没有调试成功，所以本文基于 RT1060 SDK LWIP UDP 添加组播代码并且测试。代码功能实现 RT1060 UDP 加入组播，配置本地地址，然后和 PC 端组播软件实现数据接收与发送。本文将会给出 UDP 组播原理与相关知识，在 SDK 中添加组播相关代码，给出组播配置与测试，以及 wireshark 抓包结果。

二， UDP 组播相关知识

2.1 UDP 信息传输方式

UDP 信息传递方式分为三种：

- 1) 单播 unicast: 客户端与服务端之间的点到点连接
- 2) 广播 broadcast: 主机之间“一对所有”的通信模式，广播者可以像网络中所有主机发送信息。
- 3) 组播 multicast: 主机之间“一对一组”的通信模式，加入同一组的主机可以接收到此组内所有数据。

2.2 IP 类别

IP 类别通常分为 5 大类，具体如下：

类别	第一字节 (二进制)	第一字节 取值范围	网络号个 数	主机号 个数	适用范围
A 类	0XXX XXXX	0~127	125	16777214	大型网络
B 类	10XX XXXX	128~191	16368	65534	中型网络
C 类	110X XXXX	192~223	2097152	254	小型网络
D 类	1110 XXXX	224~239	—	—	多播
E 类	1111 XXXX	240~255	—	—	保留

图 1

所以，可见本文需要用到的组播地址范围为 D 类。

2.3 组播地址范围

组播地址范围：224.0.0.0~239.255.255.255

在组播地址范围内也有 4 中类型划分。

224.0.0.0~224.0.0.255: 预留组播地址

224.0.1.0~224.0.1.255: 公用组播地址，可以用于 Internet

224.0.2.0~238.255.255.255: 用户可用组播地址 (临时组地址)，全网范围内有效

239.0.0.0~239.255.255.255: 本地管理组播地址，仅在特定本地范围内有效

所以本文选择的测试地址范围在用户可用组播地址范围内：

224.0.2.0~238.255.255.255

2.4 组播原理

组播实现一对多，组播首先由一个用户申请一个组播组，这个组播组被维护在路由器

中，其他用户申请加入组播组，这样当一个用户向组内发送消息时，网络中的路由器通过底层的 IGMP 协议自动将数据发送到所有监听这个组的终端。如果申请加入的组不足本级路由器中，而且路由器和交换机允许组播协议通过，路由器将申请加入的操作向上级路由提交。

三， RT1060 UDP 组播代码

从组播的原理，我们可以看到，组播离不开 IGMP， 还有需要申请加入一个组。本章节基于 RT1060 SDK LWIP UDP 例程添加 IGMP 相关配置与代码。

代码路径如下：

SDK_2_11_0_EVK-

MIMXRT1060\boards\evkmimxrt1060\lwip_examples\lwip_udpecho\freertos

以 MCUXpresso 工程为例：

3.1 添加 IGMP 功能

官方 SDK 已经在 SDK 的 LWIP 代码中添加的 IGMP 功能，对于用户只需要使用宏打开 IGMP 即可：

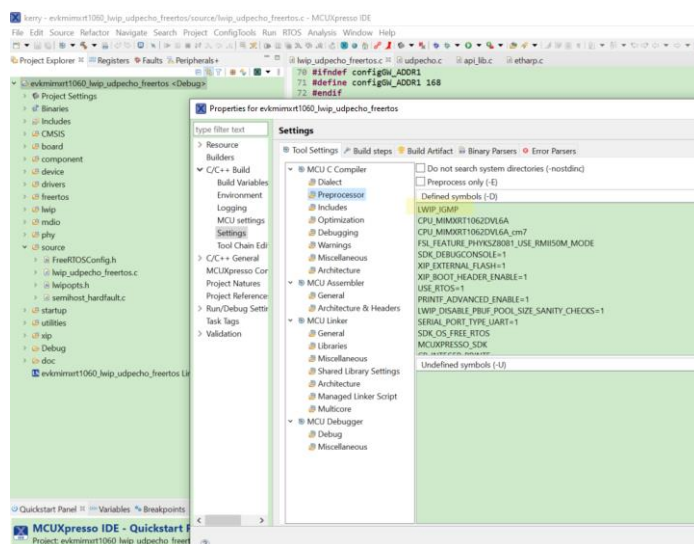


图 2

3.2 申请组播并加入

修改代码主要为 udpecho.c 中的 udpecho_thread,该函数修改后如下：

```
static void
udpecho_thread(void *arg)
{
    struct netconn *conn;
    struct netbuf *buf;
    char buffer[4096];
    err_t err;
    LWIP_UNUSED_ARG(arg);
    //kerry test
    struct ip4_addr group_addr, local_addr;
    IP4_ADDR(&group_addr, 230, 1, 1, 11); //multicast
```

```

    IP4_ADDR(&local_addr,192,168,0,102);
#if LWIP_IPV6
    conn = netconn_new(NETCONN_UDP_IPV6);
    LWIP_ERROR("udpecho: invalid conn", (conn != NULL), return;);
    netconn_bind(conn, IP6_ADDR_ANY, 7);
#else /* LWIP_IPV6 */
    conn = netconn_new(NETCONN_UDP);
    LWIP_ERROR("udpecho: invalid conn", (conn != NULL), return;);
    // netconn_bind(conn, IP_ADDR_ANY, 7);
    netconn_bind(conn, NULL, 1178);
    netconn_join_leave_group(conn,&group_addr,&local_addr,NETCONN_JOIN);
#endif /* LWIP_IPV6 */

while (1) {
    err = netconn_recv(conn, &buf);
    if (err == ERR_OK) {
        /* no need netconn_connect here, since the netbuf contains the address */
        if(netbuf_copy(buf, buffer, sizeof(buffer)) != buf->p->tot_len) {
            LWIP_DEBUGF(LWIP_DBG_ON, ("netbuf_copy failed\n"));
        } else {
            buffer[buf->p->tot_len] = '\0';
            err = netconn_send(conn, buf);
            if(err != ERR_OK) {
                LWIP_DEBUGF(LWIP_DBG_ON, ("netconn_send failed: %d\n", (int)err));
            } else {
                LWIP_DEBUGF(LWIP_DBG_ON, ("got %s\n", buffer));
            }
        }
        netbuf_delete(buf);
    }
}
}

```

主要是加入组播:

`netconn_join_leave_group(conn,&group_addr,&local_addr,NETCONN_JOIN);`
 这里端口配置为 1178,注意测试的时候选对端口。

四, RT1060 UDP 组播测试

本文测试使用 PC 端 UDP 组播工具, 实现连接组播, 发送并接收 RT 端数据。

4.1 PC 网络配置

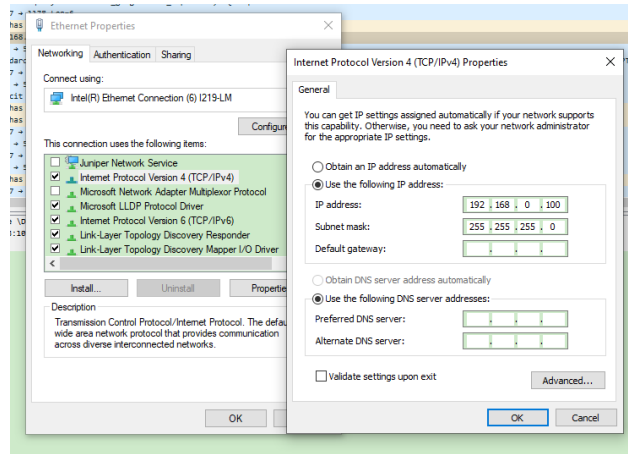


图 3

4.2 PC 端 UDP 组播软件配置

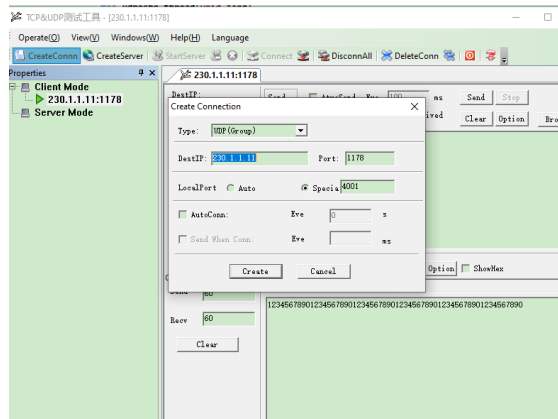


图 4

4.3 硬件连接

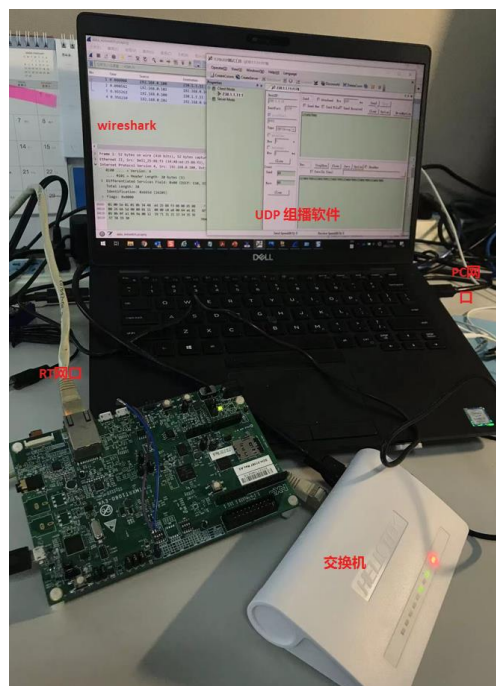


图 5

4.4 PC UDP 组播工具发送接收

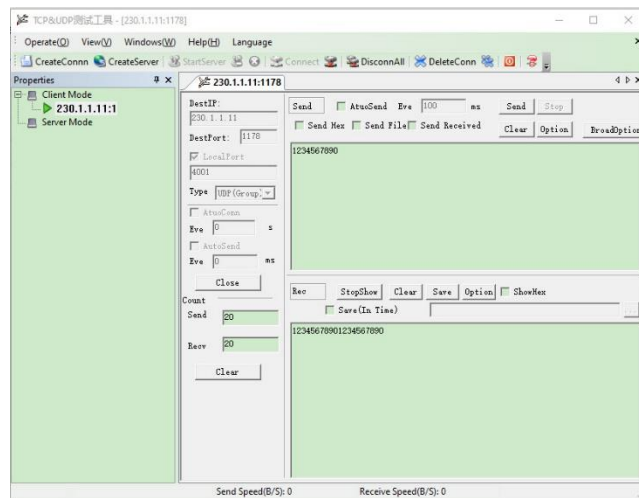


图 6

4.5 RT 打印信息

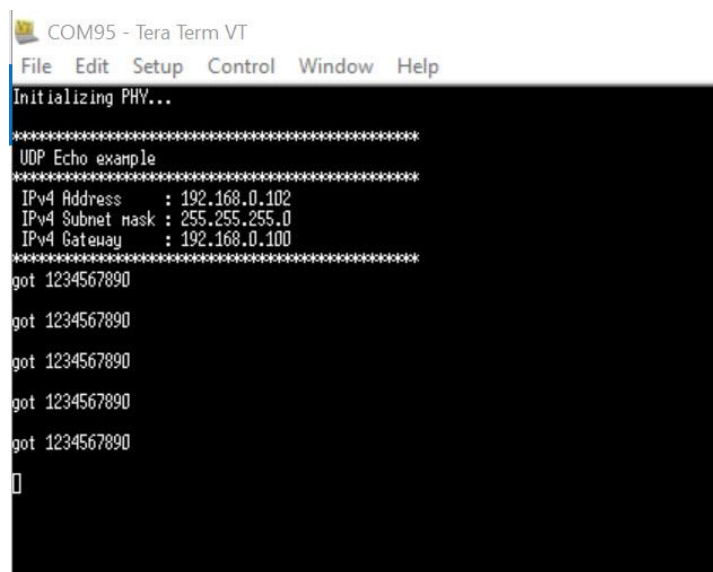


图 7

4.6 wireshark 抓包情况

下面抓包是 PC 端 UDP 组播软件发送两次的抓包情况，可以看到 PC->RT 板子是 PC IP 到多播 IP，RT 板子到 PC 是 RT local IP 到 PC IP。发送数据之后，能够成功的收到数据，所以说，本次修改代码之后，实现了 UDP 组播的功能。

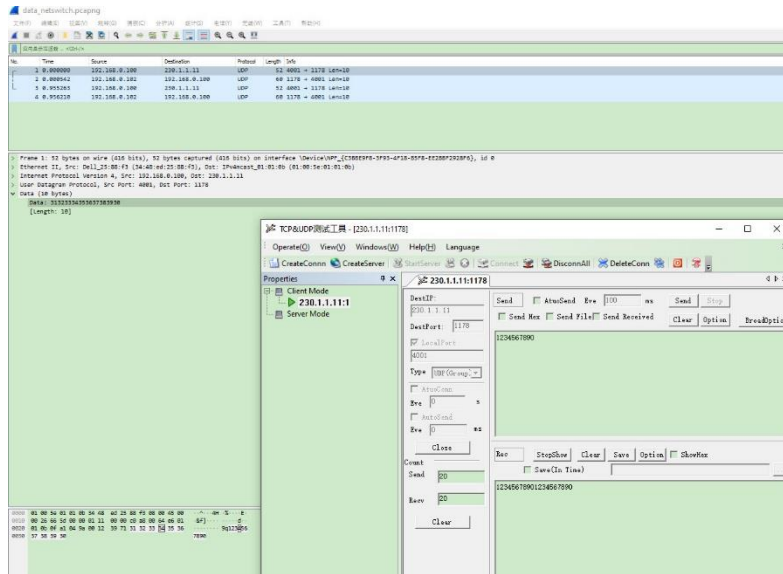


图 8

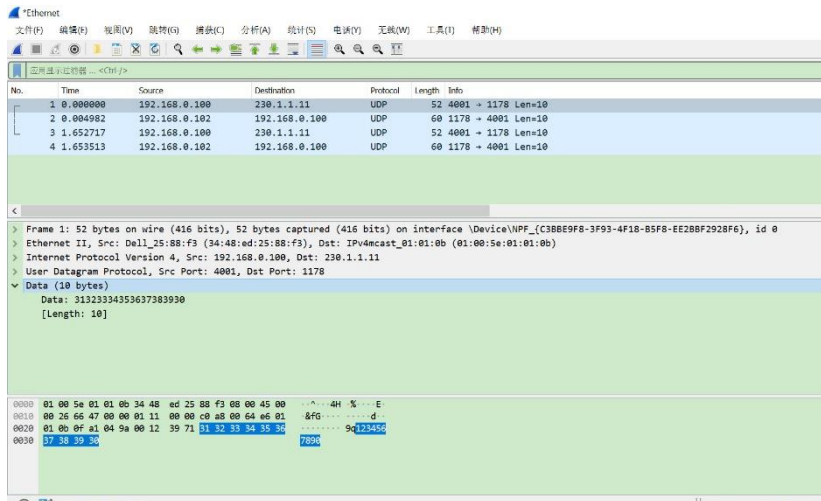


图 9

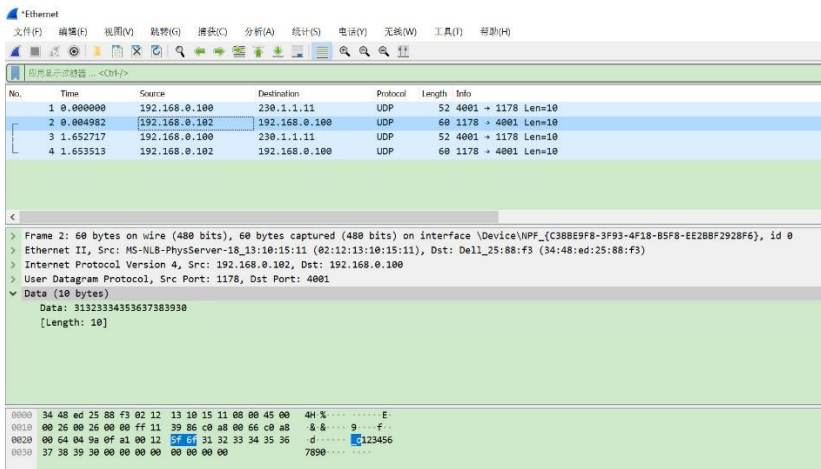


图 10