

TCP/IP 网络组件 Lwip 之 最简单的 webserver

RealTouch 评估板 RT-Thread 入门文档

版本号: 1.0.0
日期: 2012/8/27

修订记录

日期	作者	修订历史
2012/8/27	bloom5	创建文档

实验目的

- ❑ 快速了解 Lwip 网络组件。
- ❑ 了解最简单的 webserver 的工作原理和使用方式。

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

本实验在 RealTouch 上运行了一个简单的 HTTP/1.0 服务器，从而在 PC 端浏览器中可以访问 RealTouch 提供的 webserver 服务。关于 http 需要了解的相关内容可以参阅网络相关内容。

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

关于 LwIP 在 rtconfig.h 中相关宏的开启、IP 相关设置可参见上一节。

Application.c 中所做的工作与上节的 telnet 类似, 完成网络初始化, 运行 webserver 即可。

```
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. */
    webserv();
}

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;
}
```

webserver 与 telnet 一样也是作为一个线程存在的:

```
void lw_thread(void* paramter)
{
    struct netconn *conn, *newconn;
    rt_err_t err;

    /* 建立一个新的 TCP 连接句柄 */
    conn = netconn_new(NETCONN_TCP);

    /* 将连接绑定在任意的本地 IP 地址的 80 端口上 */
    netconn_bind(conn, NULL, 80);
```

```

/* 连接进入监听状态 */
netconn_listen(conn);

/* 循环处理 */
while(1)
{
    /* 接受新的连接请求 */
    err = netconn_accept(conn, &newconn);

    /* 处理进入的连接 */
    process_connection(newconn);

    /* 删除连接句柄 */
    netconn_delete(newconn);
}
}

void webserv()
{
    rt_thread_t tid;

    tid = rt_thread_create("webserv", lw_thread, RT_NULL,
        1024, 25, 5);
    if (tid != RT_NULL) rt_thread_startup(tid);
}

```

接下来，再来看看 web server 是怎么实现的。

```

const static char indexdata[] = "<html> \
    <head><title>A test page</title></head> \
    <body> \
    This is a small test page. \
    </body> \
    </html>";

const static char http_html_hdr[] = "HTTP/1.0 200
OK\r\nContent-Type: text/html\r\n\r\n";

/* 这个函数处理进入的连接 */
static void process_connection(struct netconn *conn)
{
    struct netbuf *inbuf;
    char *rq;
    rt_err_t err;
    rt_uint16_t len;

```

```

    /* 从这个连接读取数据到 inbuf, 我们假定在这个 netbuf 中包含完整的请求 */
    err = netconn_recv(conn, &inbuf);

    /* 获取指向 netbuf 中第一个数据片断的指针, 在这个数据片段里我们希望包含这个请求 */
    netbuf_data(inbuf, (void**)&rq, &len);

    /* 检查这个请求是不是 HTTP "GET /\r\n" */
    if( rq[0] == 'G' &&
        rq[1] == 'E' &&
        rq[2] == 'T' &&
        rq[3] == ' ')
    {
        /* 发送头部数据 */
        netconn_write(conn, http_html_hdr,
            sizeof(http_html_hdr), NETCONN_NOCOPY);
        /* 发送实际的 web 页面 */
        netconn_write(conn, indexdata, sizeof(indexdata),
            NETCONN_NOCOPY);

        /* 关闭连接 */
        netconn_close(conn);
    }
    netbuf_delete(inbuf);
}

```

编译调试及观察输出信息

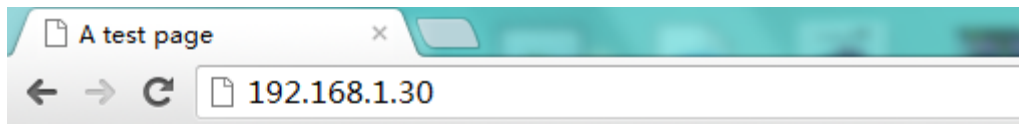
编译请参见《RT-Thread 配置开发环境指南》完成编译烧录，参考《Realtouch 开发板使用手册》完成硬件连接，连接好串口线，连上网线。运行后可以看到串口有如下的信息：

```

\ | /
- RT -   Thread Operating System
/ | \    1.1.0 build Aug 26 2012
2006 - 2012 Copyright by rt-thread team
TCP/IP initialized!
finsh>>

```

此时 RealTouch 上的 webserver 服务器也已经开启了，此时用你最最喜欢的浏览器，在地址栏内输入神圣的 192.168.1.30，登哒！！



This is a small test page.

结果分析

通过以上内容的实际操作，一个简单的 webserver 就搭建成了。