uIP协议栈下同一设备多IP的实现方法

孙建延[1],雷 钢[1,2]
(1.中州大学 工程技术学院,郑州 450044;2.郑州大学 信息工程学院,郑州 450001)

摘 要:介绍了uIP协议在嵌入式系统中的移植应用技术,在此基础上对uIP进行了修改,以适应同一设备下配置多IP地址。设置了主次IP地址,固定次IP地址,使得用户在对主IP未知的情况下也可以和设备进行通信,获取配置信息及进行参数配置。该方法拓宽了设备对网络的适应性,减少了设备生产、调试、维护中的工作强度,具有一定的推广价值。

关键词:uIP;TCP/IP移植;多IP技术
中图分类号:TP915.04 **文献标识码**:A **文章编号**:1008-3715(2010)02-0119-04

TCP/IP协议在全球互联网上取得了巨大的成功,已成为全球网络通讯的标准,将嵌入式设备引入TCP/IP接口,能充分利用其速度快、设备便宜、应用普遍等优点。由于嵌入式系统资源有限,功能针对性强,因而没有必要支持一个完整的TCP/IP协议组件,而只需要实现与需求相关的部分协议。在本文介绍的系统中,移植了免费的TCP/IP协议栈uIP,此协议栈的特点是硬件资源开销小,无须操作系统支持,使用方便。

1. 基于uIP的嵌入式TCP/IP协议栈

1.1 uIP简介

uIP是由瑞典计算机科学研究所Adam Dunkels开发的一个小型嵌入式TCP/IP协议栈,资源占用少是它的设计特点,去掉了许多全功能协议栈中不常用的功能,而保留网络通信所必要的协议机制,实现了TCP/IP协议集的四个基本协议:ARP地址解析协议、IP网际互联协议、ICMP网络控制报文协议和TCP传输控制协议。目前最新的uIP版本是uIP1.0,其体系结构如图1所示。

uIP1.0处于网络通信的中间层,其上层协议在这里被称为应用程序,而下层硬件被称为网络设备驱动。显然,uIP1.0并不是仅仅针对以太网设计的,它具有媒体无关性,事实上,uIP也带有PPP拨号支持组件。

为进一步节省资源占用,简化应用接口,uIP在内部实现上还作了如下特殊的处理:

(1)采用单一的全局数据收发缓冲区,不支持内存动态分配;

(2)基于事件驱动的应用程序接口,各并发连接采用轮循处理,仅当网络事件发生时,由uIP内核唤起应用程序处理。这样,uIP用户只须关注特定应用就可以了;

(3)应用程序主动参与部分协议栈功能的实现(如TCP的重发机制,数据包分段和流量控制),由uIP内核设置重发事件,应用程序重新生成数据提交发送,免去了大量内部缓存的占用,基于事件驱动的应用接口使得这些实现较为简单。

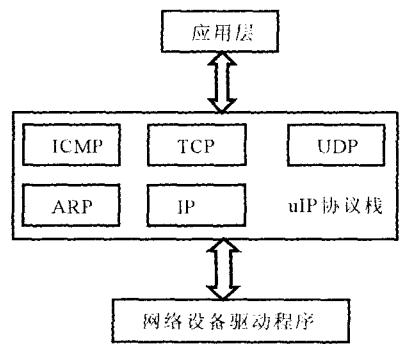

图1 uIP体系结构

1.2 uIP协议栈的移植

uIP协议通过一系列接口函数与底层系统和上层应用程序通信,它内部的协议集合对外部系统来说是透明的,从而增强了该协议的通用性和独立性,可以非常方便地移植到不同系统和应用平台。

(1)uIP的设备驱动程序接口

uIP有两个函数直接需要底层设备驱动程序的支持。一是uIP_input(),当设备驱动程序从网络层收到一个数据包时要调用这个函数,设备驱动程序必须事先将数据包存放到uIP_buf[]中,包长赋值给uIP_len,然后交由uIP_input()处理。当函数返回时,如果uIP_len不为0,则表明有网络数

收稿日期:2010-02-01
作者简介:孙建延(1974—),男,山东滨州人,硕士,中州大学工程技术学院讲师,主要研究方向:自动化装置与检测技术。

据(如SYN,正确应答等)要发送。当需要ARP支持时,还需要考虑更新ARP表或发出ARP请求和回应。另一个需要驱动程序支持的函数是uIP_periodic(),这个函数用于uIP内核对各连接的定时轮询,因此需要一个硬件支持的定时程序周期性地用它轮询各连接,一般用于检查主机是否有数据要发送,如有,则构造IP包。

(2)uIP协议栈与应用程序的接口

应用程序作为单独的模块由用户实现,uIP协议栈提供一系列接口函数供用户程序调用。用户需要将应用层入口程序作为接口提供给uIP协议栈,定义为宏UIP_APPCALL()。IP在接受到底层传来的数据包后,若需要送到上层应用程序处理,它就调用UIP_APPCALL()。

(3)以太网接口驱动

设计中采用RTL8019作为以太网通信物理层和MAC层。对于网络数据的处理,本接口软件设计提供给uIP三个和网卡相关的函数:

a)etherdev_init();//初始化函数,包括初始化RTL8019的寄存器,发送、接收FIFO设置,MAC地址等设置,此函数在RTL8019复位后仅执行一次。

b)etherdev_send();//负责数据发送处理,供uIP发送数据时调用,负责将uIP打包好的以太网数据包结构的数据发送出去。

c)etherdev_read();//负责数据接收处理,供uIP周期性调用,如果有接收到数据,则以太网数据包结构的数据存入uIP_buf[]中,并对uIP_len赋值。

(4)uIP运行流程

基于uIP的程序设计的要点是在主函数里编写循环的uIP处理,一般来说,系统的主体函数流程如下:

```
main()
{… // 变量定义及相关外设设定
uip_init(); // 初始化uIP协议栈,IP地址、子网掩码、网关,设置监听端口
etherdev_init(); // 初始化RTL8019
uip_arp_init(); // 初始化ARP表
…// 其他设置及操作
While(1)
{ …// 其他操作,以下为uIP工作循环
u8_t i; u16_t tmp;
uip_len = etherdev_read();
if(uip_len = = 0) // 没有从以太网接收到数据
{ for(i = 0; i < UIP_CONNS; i + + )
{ uip_periodic(i); // 处理周期性事件
if(uip_len > 0) { uip_arp_out(); etherdev_send();}
}
#if UIP_UDP // 如果使用UDP协议,则操作
for(i = 0; i < UIP_UDP_CONNS; i + + )
{ uip_udp_periodic(i); // 处理周期性事件
if(uip_len > 0)
{ uip_arp_out(); etherdev_send(); }
}
#endif
if( + + arptimer = = 20) // 周期性更新ARP表
{ uip_arp_timer(); arptimer = 0;}
}
else /* (uip_len ! = 0)从以太网上接收到数据 */
{ if(BUF - > type = = htons(UIP_ETHTYPE_IP))//IP包,含UDP、TCP、ICMP
{ uip_arp_ipin(); //匹对ARP表,如果没有数据发送方信息,则存起来 uip_input(); // 处理接收的数据包
if(uip_len > 0) // 要求发送数据
{ uip_arp_out(); etherdev_send(); }
}
else if(BUF - > type = = htons(UIP_ETHTYPE_ARP))// 处理ARP包
{ tmp = uip_len; uip_arp_arpin();
if(uip_len = =0) { uip_len = tmp;
static_arp_arpin(); } //为配合双IP地址而添加
if(uip_len > 0) { etherdev_send(); }
}
} // uIP工作循环结束
… // 其他操作及设置
}
}
```

(5)用户处理函数

用户接收到应用层数据包的处理函数,是在uip_input()函数中被执行的,不过在此函数中,uIP调用的是两个宏定义的函数名UIP_APPCALL和UIP_UDP_APPCALL,分别处理TCP包和UDP包,在uIP的源代码中,并未实现这两个函数。用户在移植时,需要自己来实现这两个函数。在本系统中,采用的是UDP通信,则在uipopt.h中定义如下:

```
#define UIP_UDP_APPCALL udp_data_srv
```

而udp_data_srv正是用户编写的包处理函数:

```
void udp_data_srv(void)
{if(uip_newdata()) // 如果是收到新的应用层数据
{ Process_Packet(); // 解析包内容
uip_send(); // 回发数据包给远端本地连接的IP和端口
}
/* else if (uip_connected()) 等等 */
}
```

2. 双IP实现

在实际应用中,在同一个局域网内可能要挂多个设备,所以各个设备的IP地址不同。但由于生产的需要,出厂时都设置为同一默认IP,现场安装时根据现场网络情况来设置IP(称这样配置的IP为主IP)。然而,在使用中,用户可能会忘记新设定的IP,这时给设备另置一个固定IP(称为次IP)就显得有必要了。次IP的存在,使得用户总有一条途径可以和设备进行通信,获取配置信息或进行配置参数。为

此,对 uIP 要做如下修改,以适应双 IP 同时操作。

2.1 添加响应 ARP 包的函数

仿照 uip_arp_arpin( )函数,重编写一个 Static_arp_arpin( )函数。例如定义第二个 IP 为:192.168.1.1,则

```
const u16_t static_uip_hostaddr[2] =
{HTONS((192 << 8) | 168),HTONS((1 << 8) | 1)};
```

static_arp_arpin( )函数与 uip_arp_arpin( )函数相比的差别仅在于,前者用 static_uip_hostaddr 取代了后者中的 uip_hostaddr。两者使用相同的 uip_ethaddr(即 MAC 地址)。在 uip_arp_arpin( )函数运行后,如果 uip_len = 0,则运行 static_arp_arpin( )函数。由此完成了和主机的 ARP 服务。见上文中系统主函数 main( )中流程。

2.2 在接收 IP 包函数中,增加对目标 IP 的判断条件

uIP 协议栈在 uip_input( )中会自动过滤掉不是发给本 IP 的数据,所以在这里应该做修改,即增加数据包目标 IP 的判断条件,以让协议栈能接收两个本机 IP 的数据。在 uip_process( )函数中定义:

```
#if 0 //uIP 原始代码
/* 检查数据包目标 IP 与本机 IP 是否相同 */
if(BUF->destipaddr[0] != uip_hostaddr[0]) {
UIP_STAT(++uip_stat.ip.drop);
UIP_LOG("ip: packet not for us.");
goto drop;
}
if(BUF->destipaddr[1] != uip_hostaddr[1]) {
UIP_STAT(++uip_stat.ip.drop);
UIP_LOG("ip: packet not for us.");
goto drop;
}
#else // 修改后的代码
/* 检查数据包目标 IP 是否为本机. */
if((BUF->destipaddr[0] != uip_hostaddr[0]) &&
(BUF->destipaddr[0] != static_uip_hostaddr[0]))
{UIP_STAT(++uip_stat.ip.drop);
UIP_LOG("ip: packet not for us.");
goto drop;
}
if((BUF->destipaddr[1] != uip_hostaddr[1]) &&
(BUF->destipaddr[1] != static_uip_hostaddr[1]))
{UIP_STAT(++uip_stat.ip.drop);
UIP_LOG("ip: packet not for us.");
goto drop;
}
#endif
```

2.3 构造发送数据包

修改发送包中的发送方的 IP(即源地址)信息提取方法,构造回复包的源地址,uIP 是直接读取 hostaddr 值,现直接从接收到的发送包中拷贝出来,因为前文已经对接收数据包进行了过滤,故接收到的数据包中的目标 IP 肯定是本机两个 IP 中的一个了。

```
#if 0 // uIP 代码
BUF->srcipaddr[0] = uip_hostaddr[0];
BUF->srcipaddr[1] = uip_hostaddr[1];
#else // 修改后代码
BUF->srcipaddr[0] = BUF->destipaddr[0];
BUF->srcipaddr[1] = BUF->destipaddr[1];
#endif
//构造回复包的目的地址,并去掉连接的 IP 地址判断
#if 0 // uIP 代码
if(uip_udp_conn->lport != 0 &&
UDPBUF->destport == uip_udp_conn->lport &&
(uip_udp_conn->rport == 0 ||
UDPBUF->srcport == _udp_conn->rport)) {
//&& //核对 PORT
//BUF->srcipaddr[0] == uip_udp_conn->ripaddr
[0] && //核对 IP 是否来自指定地址
//BUF->srcipaddr[1] == uip_u  _conn->ripaddr
[1]) {
goto udp_found;
#else // 修改后的代码
if(uip_udp_conn->lport != 0 &&
UDPBUF->destport == uip_udp_conn->lport) {
uip_udp_conn->rport = UDPBUF->srcport; //拷贝远
端服务器 PORT
uip_udp_conn->ripaddr[0] = BUF->srcipaddr[0];
//拷贝远端服务器 IP
uip_udp_conn->ripaddr[1] = BUF->srcipaddr[1];
goto udp_found;
#endif
}
```

3. 设备通信流程

建议在没有操作系统的系统中运行 uIP 时采用 UDP 方式而非 TCP 方式,这是因为 TCP 方式需要的网络传输开销较大,且 TCP 自带超时机制,对嵌入式系统要求较高,性价比较低,而 UDP 方式是软件开销较小,使用灵活。

通信中,一次数据包最长 1024(或更少)字节。通信总是由 PC 发起,设备被动接受连接,双方采用一问一答的形式,用超时来判断网络或通信异常,发生超时时,采用重传方式,如果重传次数过多,则认为设备未连接。设备通信流程如图 2 所示。

4. 结束语

uIP 是一个适用于嵌入式系统的 TCP/ IP 协议栈,有完整的说明文档和公开的源代码,可配置性较高,具有移植性好,配置简单,软硬件开销小等优点,可以适应不同资源条件和应用场合。文中针对实际应用需要,对 uIP 内核进行了修改,实现了一机双 IP 乃至多个 IP 的方法,这适用于同一个局域网内,同时有多台同样设备连接入网的情况。

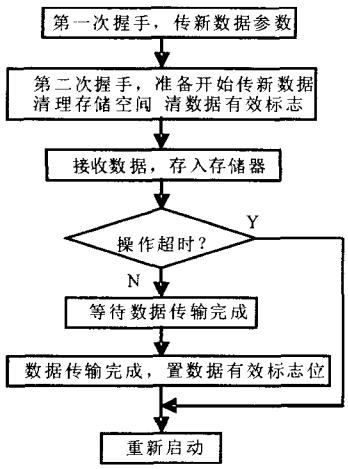

图2 网络服务程序流程

参考文献：

[1]Adam Dunkels. The uIP1.0 Reference Manual[EB/OL]. http://www.sics.se/~adam/uip/index.php/Documentation. 2006-07.

[2]AdamDunkels. uIP 源代码 uip-1.0.tar.gz[EB/OL]. http://www.sics.se/~adam/uip/index.php/Download. 2006-06.

[3]罗军舟，李波涛，杨明. TCP/IP 协议及网络编程技术[M]. 北京：清华大学出版社，2004.

[4]伊文斌，周贤娟，鄢化彪. uIP TCP/IP 协议分析及其在嵌入式系统中的应用. 计算机技术与发展[J]. 2007，17(9)：240-243.

[5]孙林，钱峰，将青. TCP/IP 协议在嵌入式操作系统 UCOS-II 中的实现[J]. 山西电子技术，2006(3)：31-33.

（责任编辑 吕志远）

Multi-IP Implementation of Same Device Based on uIP Protocol

SUN Jian-yan[1], LEI Gang[1,2]

(1. College of Engineering and Technology, Zhongzhou University, Zhengzhou 450044, China;
2. College of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: This paper introduces the transplantation technology of uIP protocol. On this basis, uIP is modified to adapt to multiple IPs on the same device. Parameters are acquired and configured by using the stationary secondary IP when the primary IP is unknown. This method is convenient for network application of devices and it reduces the work of debugging and maintaining, which is of promotional value.

Key words: uIP; TCP/IP transplantation; Muti-IP technology

（上接第 115 页）

[10]王洋，张璞，于涛，王化田，阎秀峰. 高效液相色谱法测定红景天苷含量方法的研究[J]. 植物研究，2001，21(1)：113-115.

[11]邰韧辉. 高纯度红景天苷的制备工艺研究[J]. 现代中药研究与实践，2004，18(4)：58-59.

[12]毕会敏，张守勤，刘长姣. 纤维素酶提取红景天总黄酮的研究[J]. 天然产物研究，2006，18：818-821.

（责任编辑 吕志远）

Investigation of Microwave-assisted Extraction of Salidroside

XU Xiang[1], LI Xiao-jing[1], LI Jing-jing[2]*

(1. Experiment Administration Center, Zhongzhou University;
2. College of Chemical Engineering and Food, Zhongzhou University, Zhengzhou 450044, China)

Abstract: The optimized technological parameters for extracting salidroside from rhodiola with microwave irradiation and water as solvent were studied. The key factors, including the ratio of solidto liquid, microwave power, microwave irradiation time and times of extraction, were studied for the extraction yield. The mechanism of extraction of microwave was discussed. The methods of organic flocculation, inorganic flocculation and highspeed centrifugate were applied to clarify the extract. The clarifying effect of this method is far higher than that of single flocculation method.

Key words: microwave-assisted extraction; rhodiola; salidroside; organic flocculation; inorganic flocculation