基于国产 CPU 的嵌入式医疗电子无线网络设计

传统的医疗电子设备并不具备无线功能，不能实现随时随地的医疗监控.文中提出一个基于 IPV6 的用于医疗电子无线网络的路由协议，并基于国产 CK610CPU 和 TI 公司的 CC2520 射频芯片实现了基本无线通信，并在 PC 上通过图形界面显示出医疗检验结果.

0 引言

医疗电子领域中，在嵌入式处理器方面，目前使用最广泛和主流的象 ARM.MIPS 都是国外厂商生产的 CPU,而国内的具有自主知识产权的 CPU 却很少被人注意.

此外，随着无线技术的进步和无线设备成本的降低，医疗电子无线化必将是未来发展的趋势.目前已经有一些企业和团体在医疗电子无线化的方向上作出了研究和开发，但是目前的无线医疗电子使用的网络协议基本都是基于 IPV4 的，将更加先进的 IPV6 网络运用到无线医疗电子是未来发展必须面临的问题.

本文主要为医疗电子设计了一个基于 IPV6 的网络协议，为实现该协议编写了路由协议程序和射频适配程序，实现了医疗电子的无线功能.

1 系统概述

主要系统架构如图 1 所示:单个节点由 CK610 开发板与医疗电子板和 CC2520 射频芯片组成，控制器为普通节点与 PC 相连接.Linux 用作 CK610 的操作系统，CK610 通过操作 FPGAIP 核模拟的 SPI 来控制 CC2520,CC2591 由 CC2520 的管脚控制.

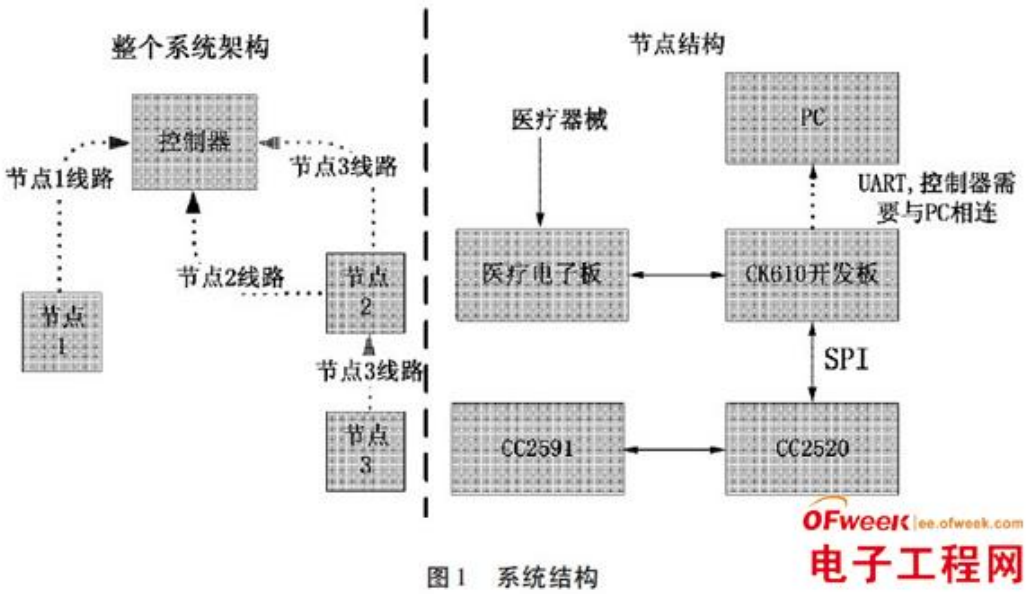

图 1　系统结构

在整个医疗电子应用的网络系统中，可以有多个节点，但是只有一个控制器. 所有节点数据发送的终点是控制器，距离较远的节点可以通过其他节点转发来传输数据.

2 网络协议设计

为了实现图 1 的系统架构，需要为系统设计一个基于 IPV6 的网络协议.

2.1 路由算法设计

建立和维护无线网络，必须要发送和处理三种形式的数据包，分别是路由请求包. 路由广播包和路由汇报包.

网络中的每个节点都维护一个路由表，路由表的每项都是与本节点相邻的节点. 一个典型节点路由表内容如表 1 所示. （点击可查看大图）

表 1　典型的节点路由表

下一跳节点（ID）	总链路代价 C	长期发送数据包总数 LT	长期发送数据包成功数 LS	近期发送数据包总数 ST	近期发送数据包成功数 SS	前驱节点链路代价 P
3	20.20	1000	988	60	58	10.08
4	20.66	1062	1024	61	57	10.20

每个节点会对路由表中的每项根据总链路代价进行排序，总链路代价最小的对应的节点会被选为默认路由，只要节点有数据要发送或者转发，都会将数据发往默认路由.

总链路代价表示的是这条链路信道的质量，该值越小越好，它是本地链路代价和前驱节点链路代价两者之和. 前驱节点链路代价是从本节点定期发送的路由广播包中获取的. 如果前驱节点是控制器，那么前驱节点链路代价是 0. 本地链路代价指的是自身节点同前驱节点之间信道的链路代价，表示的是通信信道质量，该值越小越好. 为了计算本地链路代价，必须要维护长期发送数据包总数 LT. 长期发送数据包成功数 LS. 近期发送数据包总数 ST 和近期发送数据包成功数 SS 这四个数据.

ST 和 SS 从零开始计数，节点每发送一次数据（包括重传）, 近期发送数据包总数 ST 就会加 1, 而近期发送数据包成功数 SS 则是每成功发送一次数据就加 1. 这两个值会在 RTIMER 定时器到期之时更新完长期发送数据包总数 LT 和长期发送数据包成功数 LS 之后清零. 长期发送数据包总数 LT 和长期发送数据包成功数 LS 在 RTIMER 定时器到期之时更新，更新的规则是将当前长期发送数据包总数 LT 加上近期发送数据包总数 ST 作为新的长期发送数据包总数 LT, 将长期发送数据包成功数 LS 加上近期发送数据包成功数 SS 作为新的长期发送数据包成功数 LS, 如果此时长期发送数据包总数 LT 大于 0XF000, 那么将长期发送数据包总数 LT 和长期发送数据包成功数 LS 右移一位. 这样做有两个好处：

（1）避免长期发送数据包总数 LT 和长期发送数据包成功数 LS 无限增大到无法存储.

（2）离当前时间越远的统计值对计算链路代价的影响越小，符合自然规律.

本地链路代价的计算公式如下：

$$C = 10 \times \frac{LT}{LS}$$

节点会在收到邻居节点的路由广播包和 RTIMER 定时器到期的时候更新路由表链路代价信息.

2.2 网络节点的加入

当节点 1 要加入网络中时首先要发送一个路由请求包，申请加入无线网络.控制器接收到这个请求之后，向节点 1 发送一个路由广播包.节点 1 接收到该路由广播包之后，将网络地址前缀加上自己的节点 ID 组成自己的网络地址，并且将控制器加入自己的路由表.此时节点 1 的默认路由是控制器.

同时，节点 1 会立即启动 TTIMER 和 RTIMER 两个定时器，当 TTIMER 到期时，节点 1 就会向控制器发送路由汇报包，控制器收到该包及时更新网络拓扑.

而控制器的 RTIMER 到期之时，也会给节点 1 发送路由广播，节点 1 立即更新路由表.

网络运行的过程中，有新的节点要加入这个网络，且该节点能够直接同控制器通信，此时的情况比第一个节点加入网络的情况复杂些.节点 2 启动时，首先发送一个路由请求包，控制器和节点 1 接收到该包后会先后发送路由广播包，节点 2 收到这些包之后修改自己的网络地址，并且将控制器和节点 1 都加入自己的路由表中.当 RTIMER 定时器到期时，会发送路由汇报包给控制器，经过一段时间稳定之后，路由汇报包的内容应该包括节点 1 和控制器.

之后节点 1 的 RTIMER 定时器到期，发送路由广播包，控制器和节点 2 都能收到，节点 2 更新其路由表，节点 2 的 RTIMER 定时器到期也会发送路由广播包，此时节点 1 将节点 2 加入自己的路由表.

如果新加入的节点无法直接同控制器通信，即节点 1 已经存在于网络中，节点 2 无法同控制器通信.节点 2 启动的时候也会发送路由请求包，这个包只有节点 1 可以收到，于是节点 1 给节点 2 回复一个路由广播包，节点 2 和控制器都可以收到.节点 2 收到这个包之后，会设置好自己的网络地址，同时将节点 1 加入自己的路由表中.此时节点 2 的路由表只有节点 1 这一项，因此节点 2 的默认路由是节点 1.节点 2 的 TTIMER 定时器到期时会向默认路由，即节点 1 发送路由汇报包，节点 1 收到路由汇报包之后将其转发给控制器.节点 1 的 RTIMER 定时器到期，发送路由广播包，节点 2 更新其路由表.

节点 2 的 RTIMER 定时器到期之时发送路由广播包，节点 1 接收到该包之时更新路由表.

2.3 网络的维护与更新

网络运行过程中，信道都是随时变化的.为了使整个网络工作正常，必须要及时更新网络参数.

网络的维护和更新是通过每个节点的 RTIMER 定时器实现的，每当该定时器到期，节点都会广播路由广播包，所有接收到该广播包的节点都会更新自己的路由表信息，这样整个网络的信道信息得到了更新.

3 软件和硬件设计

3.1 硬件结构

图 1 中的硬件主要包括 CK610 开发板.CC2591 和 CK610 为杭州中天微系统有限公司生产的 CPU,主要特性有：八级流水线；双发射超标量流水线技术，提升性能近 50%;非阻塞指令发射.投机执行和按序退休；返回地址预测（4－entryreturnstack）;哈佛结构数据/指令 Cache 和 SPM,大小可配置；数据 CacheWrite－back/Write－through 动态可配置；内部双通用数据总线；AHB/AXI 总线接口，和可扩展的协处理器接口.

CC2520 为 TI 公司生产的一款低功耗射频芯片，主要特性有：发射功率可达到 5dBm;数据传输速率最大可达 250kbps;工作在 2.4GHzISM 频段；4－线 SPI;6 个可配置

CC2591 为 TI 公司生产的一款高性能低成本前端，适用于如 ZigBee 网络等 2.4GHz 无线系统，可以改善 RF 性能.

CC2520 与 CC2591 结合使用可以使得输出功率范围扩大到-24dBm~22dBm,接收灵敏度增加到-90dBm,传输距离可达到几百米甚至上千米.

3.2 网络系统架构

为了实现在第 3 节中定义的网络协议,需要在 Linux 之上编写一个应用程序,该程序需要实现网络协议中的规则.除此以外，还需要根据协议，选择合适的路由，该程序称为路由协议程序.此外，为了使 IPV6 运行在 CC2520 射频芯片上，移植了 6LoWPAN 协议到 Linux 系统中,大大减小了包头的字节,减少了能量消耗.

在用户应用程序中，当有数据要发送时，Linux 内核会查找内核路由表，找到下一跳的地址，然后将数据交给相应的设备以发送数据.系统所用的网络设备是 CC2520 射频芯片，但是 Linux 并没用为 CC2520 开发对应的网络驱动，因此可以虚拟一个网络设备 tun,将网络层传过来的数据都交给 tun,读取 tun 接口便能够得到要发送的网络数据.射频适配层得到该数据之后，经过适配层 6LoWPAN 协

议的压缩之后,然后再通过SPI操作CC2520芯片发送数据.对应从底层来的数据,其处理过程跟上述方式类似.

3.3 射频适配程序的设计

当内核有数据要发送之时，要使用正确的方法驱动CC2520芯片无线发送出去，负责这部分内容的程序称为射频适配程序.

在实现过程中，使用CC2520芯片作为无线网络收发设备.在Linux内核中，并无CC2520作为网络设备的驱动.为了实现使用CC2520收发数据的功能，可以建立一个虚拟网络设备tun.字符设备tun是内核空间和用户空间的数据接口，内核将数据包发送到虚拟网络设备上，数据包会被保存到设备的队列中，此时用户空间可以通过打开字符设备tun并调用read函数读取其中的数据，此时数据传递到了用户空间，程序可以对这些数据进行处理.

同样，用户空间程序可以通过write函数将收到的数据包交给内核.这样就可以在上层有数据来的时候先经过6LoWPAN的处理压缩，再通过驱动CC2520发送数据，在CC2520收到数据的时候，将数据经过6LoWPAN还原之后交给Linux内核，实现无线收发数据包的功能.

4 系统测试

整个系统的工作流程为：医疗板首先采集医疗检验结果，将其传递给CK610开发板，开发板通过操作CC2520进行无线发送.而当CC2520接收到数据时，通过读取CC2520SO管脚获得收到的数据，通过UART输出给PC,PC运行C#编写的图形界面将检验结果显示出来，如图2所示.

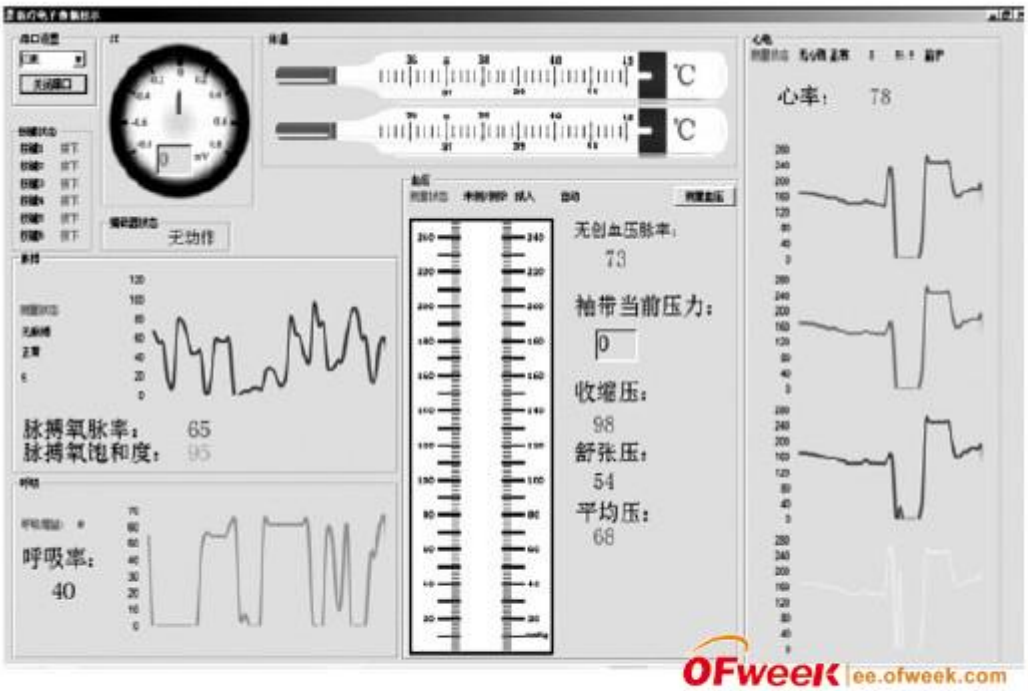

图 2　医疗电子检验结果

5 结束语

本文主要提出了一种基于国产 CPU 的 SOC 医疗电子无线网络的路由协议以及整个平台的软件架构和射频适配程序，并实现了医疗电子板的点对点无线通信，实验结果证明，系统运行正常，达到预期效果. 在后期工作中，将加入更多节点来验证系统性能.