# 基于精简协议栈的 ZigBee 网络节点研究

■重庆邮电大学 张宁
■重庆工学院 王越 王东

## 引 言

ZigBee 是一种新兴的短距离、低功耗、低数据速率、低成本、低复杂度的无线网络技术。ZigBee 在整个协议栈中处于网络层的位置，其下是由 IEEE 802.15.4 规范实现 PHY(物理层)和 MAC(媒体访问控制层)，对上 ZigBee 提供了应用层接口。

ZigBee 可以组成星形、网状、树形的网络拓扑，可用于无线传感器网络(WSN)的组网以及其他无线应用。ZigBee 工作于 2.4 GHz 的免执照频段，可以容纳高达 65 000 个节点。这些节点的功耗很低，单靠 2 节 5 号电池就可以维持工作 6～24 个月。除此之外，它还具有很高的可靠性和安全性。这些优点使基于 ZigBee 的 WSN 广泛应用于工业控制、消费性电子设备、汽车自动化、家庭和楼宇自动化、医用设备控制等。

ZigBee 协议由 ZigBee 联盟制定，是 ZigBee 的核心。目前国外带有 ZigBee 协议栈的全功能开发系统的价格非常高昂，而且 ZigBee/802.15.4 协议栈全部只提供二进制/不可修改的目标代码库供用户使用。本文研究的 ZigBee 精简版协议栈代码开放，在某些应用中可以达到标准版协议栈的效果，但是费用却低很多，因此具有较高的研究价值和应用价值。

## 1 ZigBee 精简协议栈简介

美国密西西比州立大学的 Robert Reese 教授出于教学、科研目的开发出一套精简版(subset)ZigBee 协议栈。标准协议栈和精简协议栈的功能对比如表 1 所列，可以看出，精简协议栈实现了 ZigBee 的主要功能。国内一些研究机构在此精简协议上进行扩充，实现了一些其原本不具备的功能。

这里再补充一些术语概念，这有助于理解协议栈的代码结构。

IEEE Address 节点的 8 位 802.15.4 网络地址，也称为长地址。

Network Address 节点的 2 位网络地址，也称短地址。

PAN 个人局域网。

PAN ID 个人局域网标识符。

HAL 协议栈物理抽象层。

PHY 协议栈物理层。

MAC 协议栈媒体访问控制层。

NWK 协议栈网络层。

APS 协议栈应用支持层。

APL 协议栈应用层。

精简协议栈的代码结构如表 2 所列。

表 1

| 功 能 | 标准协议栈 | 精简协议栈 |
|---|---|---|
| 路由 | 网状/树形 | 网状 |
| 网络配置 | 动态 网络发现 | 动态 网络发现 |
| PAN ID 选择 | 动态生成 | 静态 用户分配 |
| 通道选择 | 动态 | 静态 |
| 频率 | 900MHz, 2.4GHz | 2.4GHz |
| ZigBee 消息格式 | MSG | MSG |
| 直接消息发送 | 是 长/短 地址 | 是 长/短 地址 |
| 绑定 非直接消息 | 动态绑定 绑定表 | 静态绑定 绑定表 |
| 信标使能网络 | 是 | 否 |
| ZigBee 广播 | 是 | 否 |
| APS 回复 | 是 | 否 |
| ZigBee 框架 | 是 | 否 |
| ZigBee 认证 | 是 | 否 |

表 2

| 文 件 | 描 述 |
|---|---|
| msstate_lrwpan.h | 主头文件，应用程序需要包含它 |
| aps.c/h | 应用层(APL)/支持层(APS)文件 |
| nwk.c/h | 网络层(NWK)文件 |
| mac.c/h | MAC 层文件 |
| phy.c/h | 物理层文件 |
| neighbor.c/h | 邻居表，地址表函数 |
| console.c/h | 控制台输出函数 |
| debug.c/h | 调试输出函数 |
| memalloc.c/h | 堆管理 |
| ieee_lrwpan_defs.h | 对 802.15.4 的定义头文件 |
| lrwpan_config.h | 栈配置头文件 |
| lrwpan_common_types.h | 通用类型定义 |
| staticbind.c/h | 定义了存取静态绑定表的函数 |
| zep.c/h | ZigBee 端点 0 的设备函数 |
| halstack.h, evboard.h | HAL 层和评估板移植的函数原型 |

## 2 ZigBee协议编程

对于实际应用来说,最重要的是协议栈的APL函数。协议栈的每一层都有自己的有限状态机(FSM)以追踪要进行的操作。顶层的状态机函数为apsFSM(),这个函数需要最早被调用以使协议栈运行,这与标准栈中的APLTask()函数等价。所有的应用层函数都以apl或者aps开头,这些函数被分为两类:一类是对栈内数据的存取函数,一类是数据传输过程触发一系列事件的服务函数(调用)。这里需要说明的是服务调用不能重叠,这可以通过调用apsBusy()函数进行判断。

### 2.1 节点程序设计

如果节点作为协调器(coordinator),那么需要定义LRWPAN_COORDINATOR;而如果节点作为路由器(router)则需要定义LRWPAN_ROUTER;如果两者都没有定义,将作为RFD节点。

协调器节点形成网络,然后进入一个无限循环并调用apsFSM()运行协议栈。调用aplFormNetwork()服务后调用函数aplGetStatus(),如果返回了LRWPAN_SUCCESS则表示服务调用成功。代码如下:

```
main(){
    halInit();                          //初始化HAL层
    evbInit();                          //初始化评估板
    aplInit();                          //初始化协议栈
    ENABLE_GLOBAL_INTERRUPT();//开中断
    aplFormNetwork();                   //形成网络
    while(apsBusy()){apsFSM();}         //等待完成
    while(1){apsFSM();}                 //运行协议栈栈
}
```

路由器节点通过调用aplJoinNetwork()运行协议栈。代码如下:

```
main(){
    halInit();                          //初始化HAL层
    evbInit();                          //初始化评估板
    aplInit();                          //初始化协议栈
    ENABLE_GLOBAL_INTERRUPT();  //开中断
    //尝试接入网络直至成功
    do{ aplJoinNetwork();               //接入网络
        while(apsBusy()){apsFSM();}     //等待完成
      }while(aplGetStatus()!=LRWPAN_SUCCESS);
      while(1){apsFSM();}               //运行协议栈
    }
```

### 2.2 发送消息

应用程序通过调用aplSendMSG()函数发送消息包。此函数的定义如下:

```
aplSendMSG(
    BYTE dstMode,     //目标地址的地址模式
    LADDR_UNION  * dstADDR, //目的地址的指针
    BYTE dstEP,       //目标端点(直接消息方式不用)
    BYTE cluster,     //簇号(仅用于直接消息)
    BYTE scrEP,       //消息源端点
    BYTE * pload,     //用户数据缓冲区指针
    BYTE plen,        //缓冲区字节数
    BYTE tsn,         //消息的事务队列数
    BYTE reqack       //如果非0则要求确认)
```

消息从源节点的源端点发送到目标节点的目标端点。消息分直接消息(指定了目标地址)和非直接消息(仅定义了源节点、源端点和簇,没有指定目标地址)。端点号从1到255由应用程序设置(端点0由栈保留使用)。消息发送以,协议栈会向父节点路由此消息。如果收到APS的ack确认,协议栈就会将消息发送给目标端点。

### 2.3 接收消息

协议栈使用以下APL访问函数接收数据包。

| | |
|---|---|
| aplGetRxDstEp() | 返回目的端点 |
| aplGetRxCluster() | 返回簇号 |
| aplGetRxSrcEp() | 返回源端点 |
| aplGetRxSADDR() | 返回源端点的短地址 |
| aplGetRxMsgLen() | 返回消息长度 |
| aplGetRxMsgData() | 返回消息数据的指针 |
| aplGetRxRSSI() | 返回收到消息的信号强度 |

而后用户回调函数usrRxPacketCallback()将被调用。这个函数将使用用户数据结构保存数据,设置已收到数据的标志位。此函数结束后消息数据的指针将会被释放,所以在函数结束之前要将数据保存以防止下一个包将数据覆盖掉。

### 2.4 编写用户应用程序

编写用户应用程序时,要确定端点的连接方式。一种简单的方式是RFD节点周期性地向协调器节点返回数据。这样做比较简单,因为协调器的地址总是0。

RFD节点间使用直接方式通信比较困难。因为RFD节点的短地址是由其接入网络的顺序和深度决定的,事先并不知道。当然可以在协调器节点上增加程序告知RFD节点它们的地址,但这使复杂程度增加了。比较好的方式是使用非直接消息方式进行RFD节点间通信。RFD节点都将消息发送给协调器节点,协调器节点根据绑定表向正确的节点发送数据。

整个程序的运转是靠一个有限状态机维持的。图1给出了这个状态机的状态转移图。

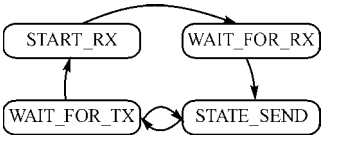


图1 有限状态机状态转移图

## 2.5　函数总结

鉴于APL层函数接口对程序设计的重要性，将这些函数做一个总结。

表3　APL服务调用

| 函　数 | 功　能 |
|---|---|
| aplSendMSG() | 发送消息 |
| aplSendEndDeviceAnnounce() | 宣告网络接入协调器 |
| aplPingParent() | 查询到父节点的连接状态 |
| aplFormNetwork() | 协调器形成网络 |
| aplJoinNetwork() | 尝试接入网络 |
| aplRejoinNetwork() | 尝试在此接入父节点 |

表4　APL/APS访问和功能函数

| 函　数 | 功　能 |
|---|---|
| aplShutdown() | 关闭协议栈 |
| aplWarmstart() | 唤醒协议栈 |
| apsBusy() | 栈忙则返回非0 |
| aplMacTicksToUs() | 转MAC滴答为$\mu$s |
| aplInit() | 协议栈初始化 |
| apsGenTSN() | 产生事务队列号 |
| aplGetMyShortAddress() | 返回节点短地址 |
| aplGetParentLongAddress() | 返回父节点长地址 |
| aplGetParentShortAddress() | 返回父节点短地址 |
| aplSetMacMaxFrameRetries(x) | 设置MAC帧最大值 |
| aplSetApsMaxFrameRetries(x) | 设置APS帧最大值 |
| aplIsUsrBufferFree() | 最后一个消息被拷贝则非0 |

表3是APL服务，这些函数需要调用apsBusy()确定其是否完成，并且使用aplGetStatus()函数返回状态。表4是APL/APS访问及功能函数。

## 结　语

无线传感器网络具有广阔的应用前景，由ZigBee协议可以方便有效地组建无线传感器网络。在整个应用中，主要硬件设备可由一个51单片机加上2.4 GHz的收发模块组成，采用CC2430是为了更加方便使用，而ZigBee的真正核心是安装在单片机中的协议栈代码。精简版协议栈不论从开发难度到使用成本都具有一定的优势。本文对精简版协议栈尤其是应用层接口、代码实现进行了详细的分析，并以此为基础给出了节点的软、硬件设计。了解协议栈的使用，就可以在其上开发适合我们需要的各种应用。

参考文献

[1] Reese Robert B. A Zigbee-subset/IEEE 802.15.4 Multi-platform Protocol Stack V0.1[2008-08-20] www.reesemicro.com.

[2] 孙利民，李建中，陈渝，等. 无线传感器网络[M]. 北京：清华大学出版社，2005.

[3] TI. SmartRF CC2430 Peliminary(rev.1.01), 2005.

（收稿日期：2008-09-10）

▶75

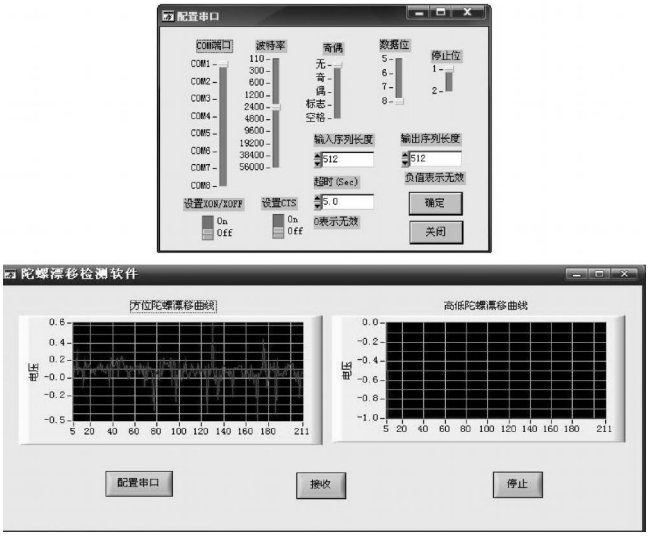


图5　用户界面

## 结　语

本文设计的数据采集系统已经得到实际应用。新型DSC拥有高精度浮点运算能力，其内部的总线结构和指令算法非常适合于数字信号处理变换，使系统采集的数据更加稳定、准确和高效。早已成为PC标准的通用串行总线(USB)则为数据的采集和传递提供了很大的便利。在开发该系统时，使用LabWindows/CVI软件可以方便地调用硬件驱动程序及功能函数，降低了软件开发工作量，加快了系统的开发周期。同时也减少了测量周期，使数据传输速度得到提高。

参考文献

[1] 罗锦，孟晨，杨锁昌. 通用自动测试平台研究[J]，中国测试技术，2005，31(5)：9-12.

[2] 刘君华. 虚拟仪器编程语言LabWindows/CVI教程[M]. 北京：电子工业出版社，2001.

[3] 章小梅，姜茂仁，郑海波. 基于LabWindows/CVI平台数字信号处理实验软件的研制[J]. 仪器仪表学报，2001，22.

[4] 宋宇峰. LabWindows/CVI逐步深入与开发实例[M]. 北京：机械工业出版社，2003.

吕强(教授、博士生导师)，研究领域为坦克火控系统、鲁棒控制理论研究与应用、机器人控制。

（收稿日期：2008-09-24）