

Z-Stack 剖析及其在无线测温网络中的应用^①

Application of Z-Stack in ZigBee Thermometric Network

张奇松 尹 航 (杭州电子科技大学 智能控制与机器人研究所 浙江 杭州 310018)

摘要: 针对无线传感器网络在应用中的能耗问题, 提出将 ZigBee 技术应用于无线传感器网络, 采用超低功耗单片机和无线模块构成节点的方案, 在剖析德州公司发布的免费协议栈 Z-Stack 基础上, 架构了应用于无线温度监测的超低功耗 ZigBee 网络, 可满足无线监测系统需长时间工作的要求, 具有很好的应用前景。

关键词: 低功耗 低速率 温度监测 无线传感器网络 无线通信协议

无线传感器网络通过在区域内部署大量的廉价微型传感器节点, 以其分布式和自组织的特点带来了信息感知的一场变革。在其应用过程中也存在着一些问题, 如无线节点的电池寿命短, 要经常人为地进行电源维护。ZigBee 作为一种专注于低功耗、低成本、低速率的无线网络通信标准, 是近年来无线通信领域研究的热点, 被广泛地应用于无线传感器网络如家庭控制、楼宇自动化和工业自动化等。

ZigBee 联盟选择了 IEEE802.15.4^[1] 作为 MAC 层和 PHY 层标准, 并于 2006 年 12 月发布了 06 版的网络层及更高层的规范^[2]。各厂家纷纷开发各自的 ZigBee 协议栈软件, 其中 Microchip 发布的协议栈对硬件的支持面过于狭隘且不支持路由器模式; FREESCALE 的协议栈工具 BEE KIT 和 EMBERR 公司的 ZIGBEE 2006 价格昂贵。以上种种因素限制了 ZigBee 协议栈在开发研究中的推广和应用。

2007 年 4 月, 德州仪器推出业界领先的 ZigBee 协议栈 (Z-Stack) 的免费下载版本^[3]。Z-Stack 符合 ZigBee 2006 规范, 支持多种平台, 包括基于 CC2420 收发器以及 TI MSP430 超低功耗单片机的平台。Z-Stack 包含了网状网络拓扑的几近于全功能的协议栈, 在竞争激烈的 ZigBee 领域占有重要地位。本设计将应用 Z-Stack 构建一个新型无线测温网络, 以实现无线传感器网络节点在电池供电下的长时间运行。

1 IEEE 802.15.4和ZigBee联盟

ZigBee 协议栈模型如图 1 所示。ZigBee 的栈结构是由多个层构成的, 各层为其上层提供了一组服务: 数据实体提供数据传输服务, 管理实体提供其它服务。每个实体都开了一个通向高层的服务接入点开进行通讯的, 如图 1 中, xxDE-SAP 是数据实体的服务接入点, xxME-SAP 为管理实体的服务接入点, 各层之间通过数据和管理实体接入点协同工作, 共同完成 ZigBee 节点的运作。

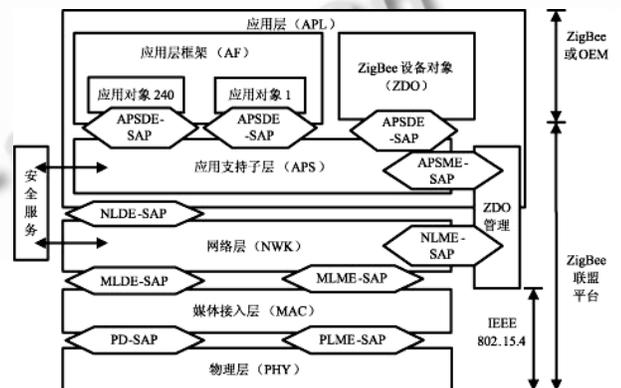


图 1 ZigBee 协议栈模型

IEEE 802.15.4-2003 标准定义了 ZigBee 协议栈的底下两层: 物理层与媒体接入层。物理层共定义了 27 个信道, 其中 2.4~2.4835GHz 包含 16 个传输速度为 250kb/s 的信道, 是全球免费的 ISM 频段;

① 基金项目:浙江省新苗人才计划项目(2007R40G2040068)

收稿时间:2008-08-20

媒体接入层利用载波监听多路访问/冲突防止(CSMA/CA)机制来控制信道接入。物理层与媒体接入层的详细内容参见文献[1]。

在物理层和媒体接入层基础上,ZigBee 联盟制定了网络层和应用层的标准。其中网络层的功能包括:加入/离开网络、安全、路由、发现单步邻居和存储邻居信息等^[4]。应用层包括应用支持子层、应用层框架和 ZigBee 设备对象。这里重点提一下应用层框架。它是 ZigBee 设备中所有应用对象的活动环境,在这个框架中可以定义 240 个独立的应用对象。ZigBee 用配置文件(Profile)对一些特定的应用对象进行了框架规范,如家庭控制照明^[5]。在后文的应用中,主要针对自定义的应用层测温对象展开。

2 Z-Stack协议栈

2.1 总体软件框架

Z-Stack 协议栈的源码总体上由三部分构成:硬件抽象层、操作系统抽象层和 ZigBee 协议各层。由于协议栈各层的内容都以任务事件的形式被集成,因此从程序执行角度来讲,主要接触的是前面两个部分。Z-Stack 协议栈的总体框架如图 2 所示。



图 2 Z-Stack 协议栈总体框架

硬件抽象层给系统应用提供了定时器、通用 IO 口、UART、ADC 的 API。它是一个独立平台的 API,提供各种服务的扩展集。在本系统中它是基于 CC2420 加 MSP430FG4619 平台的 API,并不适用其它的硬件平台。

操作系统抽象层都是以 C 语言源码的形式给出,在应用中基本不用修改(除了任务初始化函数 `osalAddTasks`,该函数在应用层中修改)。它提供了 Z-stack 使用者一个易用的操作系统平台,所有 ZigBee 的应用都可以基于此系统之上以任务的形式完成。

在系统平台上,系统任务和应用任务中的事件依时间片进行轮转。在系统级 OSAL 的时间片轮转函数中设置任务的调度,应用事件将任务划分为多个块,每个时间片内执行一个块,这样对一个时间片来说其忙碌时间就不会过长。

2.2 节点低功耗实现

Z-Stack 提供了功耗管理,这一点充分体现了 ZigBee 的低功耗特点。Z-Stack 提供了两种睡眠模式:轻度睡眠和深度睡眠。当有活动需要在延迟一段时间之后激活,那么使用轻度睡眠;如果没有任何预定的活动,那么就进入深度睡眠,只有外部的激励(比如按键动作)才能将该设备唤醒。轻度睡眠大概可以将功耗降为 mA 级,而深度睡眠则降为 uA 级。

在本系统的应用中,终端节点(含传感器)按照一定周期唤醒来处理发送数据,远端的控制器设备只有当有按键按下才是才唤醒。这些设备最主要的特点就是它们的大部分时间都是处在睡眠模式,使电池的使用时间达到最长。

判断是否进入省电模式是在主程序循环体的最后项进行的。如果没有什么任务等着去执行,Z-Stack 还要确认两个条件:(1)确定该设备是由电池供电的设备;(2)确定没有任务将睡眠功能挂起。以上两个条件满足,那么就进入省电模式。如果下一个到期时间大于零且小于 `MIN_SLEEP_TIME`,那么进入轻度睡眠模式,如果下一个到期时间为零,则进入深度睡眠。

3 基于 ZigBee 测温系统的实现

一些诸如储存药品的仓库对室内温度要求非常苛刻,由于仓库本身空间大,再加上空气流动等因素的影响,导致无法保证各分区的温度维持在恒定水平,对货物造成影响甚至损失。ZigBee 无线通讯协议的低功耗特性可以满足无线节点长期工作的需求,其低速率特性可以满足对温度信号的传输,而其低成本及免执照频道的特性也是生产应用的保障。

ZigBee 测温系统的节点硬件结构如图 3 所示。MSP430 是德州仪器的超低功耗的 16 位单片机,CC2420 是 Chipcon 公司(已被 TI 收购)推出的一款符合 IEEE 802.15.4 规范的 2.4GHz 射频芯片。温度传感器每隔一秒钟将检测到的温度信号送入 MSP430 的 A/D 采样通道,MSP430 将获得的数据通过 SPI 传输给 CC2420 无线模块。CC2420 不单负责收发温度信号,还有网络建立、网络维护等等信号。其中虚线部分硬件在除了协调器以外的节点是不存在的,按钮用来选取该显示哪个节点的温度,显示部分由一片定制的字段式 LCD 构成。协调器将数据存储并通过接口与 PC 终端通讯。

由 ZigBee 规范可知,一个节点可以有多个应用对象,而每个应用对象只能支持一个配置文件。本系统的应用对象中最重要的是周期温度监测对象,针对该对象在应用层建立了一个任务: `TemperApp_ProcessEvent`,功能如下:(1)温度检测数据传输对象,负责将终端设备的温度检测信号按

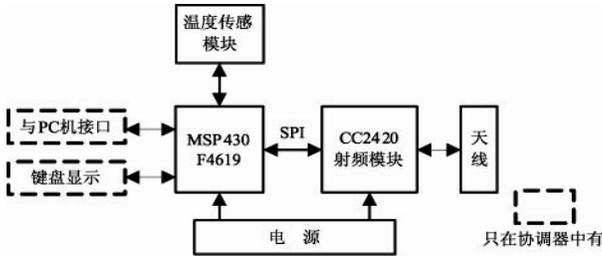


图 3 节点硬件结构图

周期发送到协调器；(2)按键对象的处理柄，根据按键来显示不同节点的温度数据。

另外，ZigBee 协议栈其它各层的工作也以独立任务的形式被申明和加入(由函数 `osalTaskAdd` 实现)，ZigBee 协议栈与 Z-Stack 任务函数的对应关系可以如图 4 所示，同一虚框内的内容相互对应。其中除应用层框架以外的代码由 Z-Stack 提供。

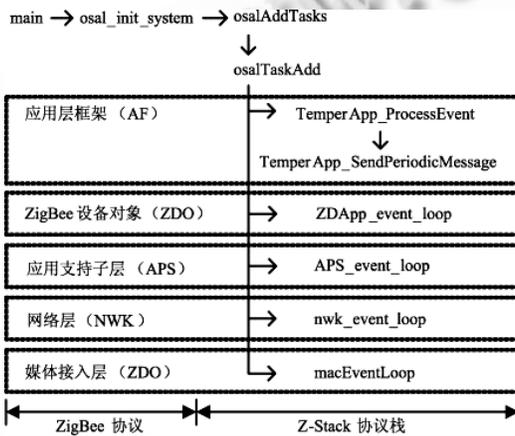


图 4 ZigBee 协议栈与 Z-Stack 任务函数的对应关系

在节点调试过程中，利用 Z-Stack 软件包携带的 Z-Tool 工具，可通过 RS-232 与 ZigBee 硬件节点进行通讯。协调器建立网络后，从上位机发送一个 `SYS_GET_DEVICE_INFO` (获得设备信息) 的系统命令，协调器将返回给上位机内容如图 5 (a) 所示。其中 `IEEEAddress` 是全球唯一的 64 位网络地址；`ShortAddress` 是 16 位的物理地址，协调器的物理地址为 0；`DeviceType` 是设备类别，3 表示该设备是全功能类型；`DeviceState` 是设备状态，表示该设备是以协调器的身份启动的；`NumAssocDevices` 为与该设备连接的设备数，此时为 0。再打开一个温度检测子节点，如图 5 (b) 所示，可看到 `NumAssocDevices` 立即变成了 1，表示有新设备与该协调器连接，温度检测网络组网成功。

系统正常运行时，协调器节点的 LCD 屏上会按周

```
<Tx> [21:39:24:187] SYS_GET_DEVICE_INFO
<Rx> [21:39:24:218] SYS_GET_DEVICE_INFO_RESPONSE
{
  Status = 0x0
  IEEEAddress = 0x108D8815F2E3004A
  ShortAddress = 0x0
  DeviceType = 0x3
  DeviceState = 0x9
  NumAssocDevices = 0x0
```

(a)

```
<Tx> [21:41:48:203] SYS_GET_DEVICE_INFO
<Rx> [21:41:48:218] SYS_GET_DEVICE_INFO_RESPONSE
{
  Status = 0x0
  IEEEAddress = 0x108D8815F2E3004A
  ShortAddress = 0x0
  DeviceType = 0x3
  DeviceState = 0x9
  NumAssocDevices = 0x1
```

(b)

图 5 子节点加入前后返回的设备信息对比

期显示来自其它节点的温度参数，达到了监测环境温度的目的。在本系统中，终节点承担的主要任务是周期性的，即将温度信号周期性的传送到协调器，由于传输数据量很小，终节点绝大部分时间是处在睡眠模式，实现了本系统的低功耗。

4 结束语

基于 ZigBee 的温度监测系统由于其低功耗、低成本的特点将会在温度监测领域有很广阔的应用前景。试验证明，使用 Z-Stack 协议栈在 CC2420+MSP430 硬件平台上开发的温度检测系统具有很好的稳定性和实用性，能够满足生产生活过程中对温度长期监测的功能需求。在其后的研究中将扩大网络规模，并研究因此而带来的网络的链路损耗问题，选取适当的路由算法并加以改进。

参考文献

- 1 IEEE Std. 802.15.4-2003. Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Low Rate Wireless Personal Area Networks[S].
- 2 ZigBee Alliance. ZigBee Specification. <http://www.zigbee.org>. 2006-12-01.
- 3 Texas Instruments Inc. Z-Stack™ Version 1.4.2. <http://www.ti.com/zigbee>. 2007-04-31.
- 4 李文仲,段朝玉. ZigBee 无线网络技术入门与实战. 北京:北京航空航天大学出版社, 2007: 56-61.
- 5 Zucatto FL, Biscassi CA. ZigBee for Building Control Wireless Sensor Networks. SBMO/IEEE MTT-S IMOC, 2007: 511-515.