

# Design and Research of Internet of Things in Environmental Monitoring Based on ZigBee\*

Hui Wang, Sai Shen

Department of Electronic Engineering, Hubei University of Economics, Wuhan  
Email: wanghui\_one@126.com, 309783127@qq.com

Received: Oct. 8<sup>th</sup>, 2013; revised: Oct. 12<sup>th</sup>, 2013; accepted: Oct. 18<sup>th</sup>, 2013

Copyright © 2013 Hui Wang, Sai Shen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract:** With the development of society, people pay more attention to the quality of lives, especially the environmental conditions. The ZigBee-based Internet of Things in environmental monitoring is designed and discussed in this paper. From the perspectives of technical research and practical application, the paper elaborates the design and realization of the application layer on the TI's CC2530 by exploring Z-Stack protocol. Furthermore, this paper describes the solution for environmental monitoring based on ZigBee, which implements a variety of functions including data collection, data transmission, data processing, intelligent terminal view. The designed system has the following features: low complexity, low power consumption and ease of deployment which have some application values in prevalence of ZigBee-based environmental monitoring.

**Keywords:** ZigBee; Z-Stack; Environmental Monitoring; Internet of Things; Sensor

## 基于 ZigBee 的环境检测的物联网的设计与研究\*

王 挥, 申 赛

湖北经济学院, 电子工程系, 武汉  
Email: wanghui\_one@126.com, 309783127@qq.com

收稿日期: 2013 年 10 月 8 日; 修回日期: 2013 年 10 月 12 日; 录用日期: 2013 年 10 月 18 日

**摘 要:** 随着社会的发展, 人们对生活质量关注度越来越高, 特别是对环境的关注。针对环境监测问题, 结合 ZigBee 技术, 本文进行了基于 ZigBee 的环境监测的物联网的设计与研究。文章从技术研究与实际应用的角度出发, 在深入分析了 Z-Stack 协议的基础上, 详细阐述了基于 TI 公司提供的 CC2530 的开发平台应用层的设计与实现, 并给出了一个基于 ZigBee 环境监测的实现方案, 整套方案实现了使用传感器进行数据采集, 节点传输, 数据处理, 以及智能终端查看的功能。本文设计的环境监测系统具有低复杂度、低功耗、易于部署等特点, 对基于 ZigBee 的环境监测的普及具有一定的应用价值。

**关键词:** ZigBee; Z-Stack; 环境监测; 物联网; 传感器

### 1. 引言

21 世纪是信息化的世纪, 各种电信和互联网新技

术推动了人类文明的巨大进步。舒适、良好的生活环境是人们追求的目标, 环境监测系统能够使人们使用高科技带来的简便而时尚的现代生活。本文设计和实现的面向环境监测的系统可以通过手机或其它智能设备检测给定区域的环境信息。此外, 一旦环境出现

\*基金项目: 本课题得到 2012 年国家级大学生创新创业训练计划项目 (No. 201211600011), 湖北省教育厅项目 (B2013032), 湖北经济学院青年基金项目 (XJ201116 和 XJ201305) 的资助。

异常时,该系统还提供多途径报警、远程监控等功能,可以及时发出警报,提醒周围的人在第一时间进一步采取行动。

本系统采用基于 CC2530 的 MCU 作为主控制器。单片机应用系统由硬件和软件组成。硬件由实现单片机系统控制要求的接口电路和有关的外围电路芯片或部件以及各种传感器组成,软件是基于 TI 公司的 Z-Stack 协议完成的<sup>[1]</sup>。在单片机应用系统开发的过程中,应不断调整软、硬件,协调地进行软、硬件设计,以提高工作效率,当系统硬件和软件紧密配合、协调一致,就可以组成高性能的单片机应用系统。本文完成了单片机应用系统其开发过程的系统的总体设计、硬件设计、软件设计和系统调试。设计的 MCU 与各个芯片和模块的接口、各项标准都严格遵循国家有关标准,为以后的产品化提供了良好的基础<sup>[2]</sup>。

在设计本系统时,面对各种检测对象和大量控制单元,需要利用各种接口标准和 MCU 进行连接,再经过 MCU 进行数据处理,实现实时测控。系统采用单片机来实现环境监测系统不仅具有采集控制方便、简单、灵活等优点,而且可以大幅度提高各模块和芯片的协调性,从而大大提高系统的可利用性。本系统也可应用于工农业中,实现对无人值守岗位的远程监控等。

## 2. 方案设计

在总体方案设计中,确定单片机控制系统是进行

系统设计最重要、最关键的一步。总体方案的好坏,直接影响整个控制系统的性能及实施细则。总体方案的设计主要是根据被控对象的任务及工艺要求而确定的。设计关键过程如下:根据系统的要求,首先确定出系统是采用开环系统还是闭环系统,或者是数据处理系统。选择检测元件,在确定总体方案时,必须选择好被测参数的测量元件,它是影响控制系统精度的重要因素之一。选择执行机构,执行机构是控制系统的重要组成部分之一。执行机构的选择一方面要与控制算法匹配,另一方面要根据被控对象的实际情况确定。选择输入/输出通道及外围设备。选择时应考虑以下几个问题:被控对象参数的数量;各输入/输出通道是串行操作还是并行操作;各通道数据的传递速率;各通道数据的字长及选择位数;整个系统流程图和原理图如图 1 所示。该系统采用树状型网络结构,首先在被监测的环境,根据具体环境布置多个合适的节点数,每个节点跟环境的实际需求搭建合适的传感器电路;每个节点将采集的数据周期性传输给路由器,路由器的布点则要根据数据传输的需求,并且考虑路由器出问题后,是否有替代的路由器进行布点;最后路由器将自己接收到的数据传输给协调器,协调器的位置作为总接收点,要能接收到各个路由器传来的数据;协调器则与 PC 机相连,通过串口将接受到的数据传输到 PC 机上的数据处理的服务器,并将其根据本地时间存储在数据库中;除节点用电池供电外,其它均用稳定电源供电。

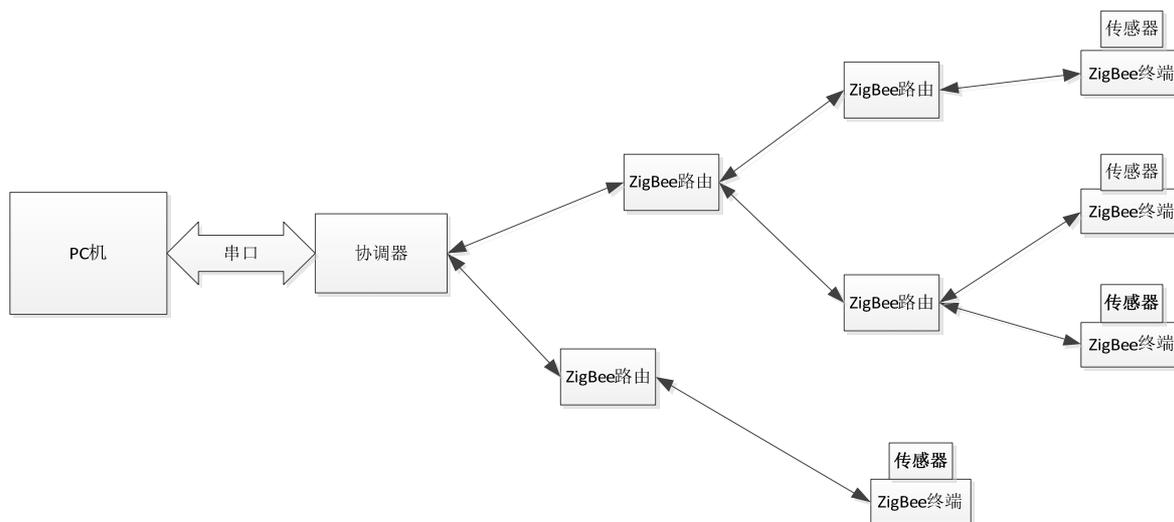


Figure 1. System Structure  
图 1. 系统结构框图

### 3. 硬件电路设计

系统硬件设计主要采集节点传感硬件设计、路由模块的硬件设计等。本文以 TI 公司提供的增强型 CC2530 单片机为核心，构成具备数据采集，数据通信等功能的硬件系统。采集节点主要是依赖传感器将环境中的各种因素转换为电信号，根据需要将其进行放大，应进行 A/D 转换，最后将其传输给采集节点的单片机。各采集节点通过 ZigBee 无线通信网络实现梳妝网络设计。PC 机通过 RS232 与 CC2530 实现串口通信<sup>[3]</sup>。

#### 3.1. 采集节点硬件设计

采集节点硬件主要有 CC2530 无线单片机和各种传感器(温度传感器、湿度传感器、烟雾传感器等等)以及电源模块构成，如图 2 所示。采集节点采用的是 TI 公司提供的 ZigBee 新一代 SOC 芯片 CC2530。CC2530 是针对 2.4 GHz 频带、使用 IEEE802.15.4 标准的系统解决方案。由于 CC2530 具有不同的运行模式，使得它具有能运行超低功耗的系统。因此，采集节点仅需普通的 5 号电池就可以工作 6~24 个月。其内置 ZigBee 协议，占用存储空间小，极大的降低了 ZigBee 成本，并且其协议是免专利费的<sup>[4]</sup>。

根据监测环境状况需要，本文选用温度传感器 DS18B20 作为代表介绍。DS18B20 数字温度传感器接线方便，封装后可应用于多种场合，如管道式，螺纹式，磁铁吸附式，不锈钢封装式，型号多种多样，有 LTM8877，LTM8874 等等。根据应用场合的不同而改变其外观。封装后的 DS18B20 可用于电缆沟测温，高炉水循环测温，锅炉测温，机房测温，农业大棚测温，洁净室测温，弹药库测温等各种非极限温度场合。耐磨耐碰，体积小，使用方便，封装形式多样，适用于各种狭小空间设备数字测温和控制领域。如图 3 是温度传感器的电路。

#### 3.2. 路由器和协调器硬件设计

协调器和路由器的硬件设计大部分是相同的，都是采用 CC530 单片机为电路的核心。但是协调器会通过串口将接收到的数据发送的 PC 机上。如图 4 和图 5 为协调器和路由器结构框图。

### 4. 系统软件设计

整个系统软件分为三部分。底层协议、服务器数据处理和智能终端客户端。

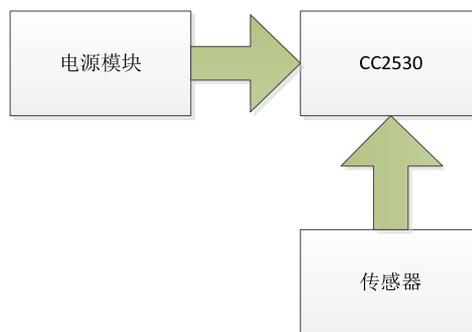


Figure 2. Terminal Node  
图 2. 终端节点框图

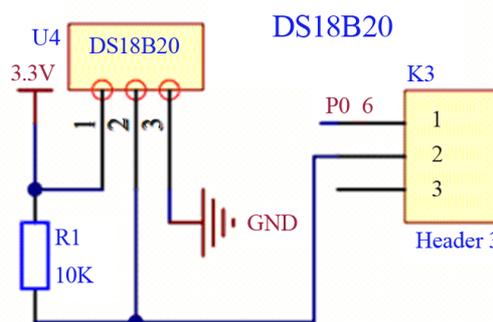


Figure 3. Circuit Diagram of Temperature Sensor  
图 3. 温度传感器电路图

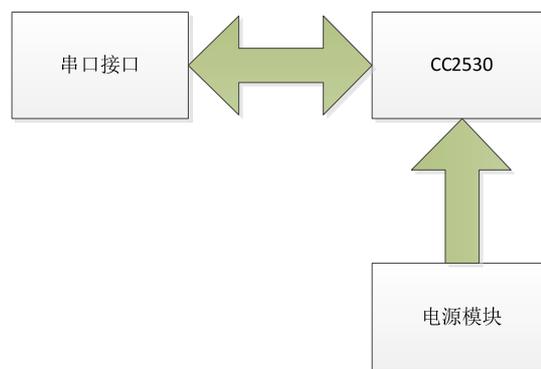


Figure 4. Structure of coordinator  
图 4. 协调器结构框图

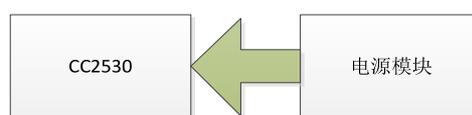


Figure 5. Structure of router  
图 5. 路由器结构框图

TI 公司提供的 Z-Stack 协议栈基于一个操作系统而存在的，它类似 uCOS 操作系统。和所有的操作系统启动一样，需要完成系统的初始化，为 OS 运行做好准备，主要有系统时钟的初始化，检测芯片的工作电压，初始化与配置系统相关的定时器，初始化芯片的各个硬件模块，初始化 NV 变量等。这些初始化完成后，就开始进入任务循环，通过比较任务的 ID 号，然后调用相应的事件处理函数进行相应的处理。如图 6 是协调器、路由器和终端传感器节点的系统运行图。

本文采用的底层协议是在 TI 公司提供的 Z-Stack 协议基础上，结合实际硬件电路进行设计，它可以兼容 IEEE 802.15.4/ZigBee 的 CC2530 片上系统解决方案。整个协议采用 C 语言编写，它结果清晰，移植方便，层次分明，可以在多任务的环境中运行，其在硬件层面上分为协调器、路由器和采集节点 3 种类型<sup>[5]</sup>。

此协议采用的是 IAR Embedded 的开发环境，三种不同的硬件类型可根据需要进行相应的选择。

Z-stack 协议栈的分成结构如下：

**App:** 应用层目录，用户在这个区域可以创建各种不同的工程，这个目录有应用层的内容及该项目的内容，在协议栈中是基于 OS 的任务实现的。用户可根据需要添加相应的文件。

**HAL:** 硬件层目录，该目录包含与硬件相关的驱动、配置及操作函数。

**MAC:** MAC 层目录，MAC 层可分为高层和底层，里面含有 MAC 层参数的配置文件和 MAC 层库的函数接口文件。

**MT:** 监视测试目录，主要用于调试，完成串口命令的收发解析，把消息发给 OS，然后 OS 对它们进行处理。

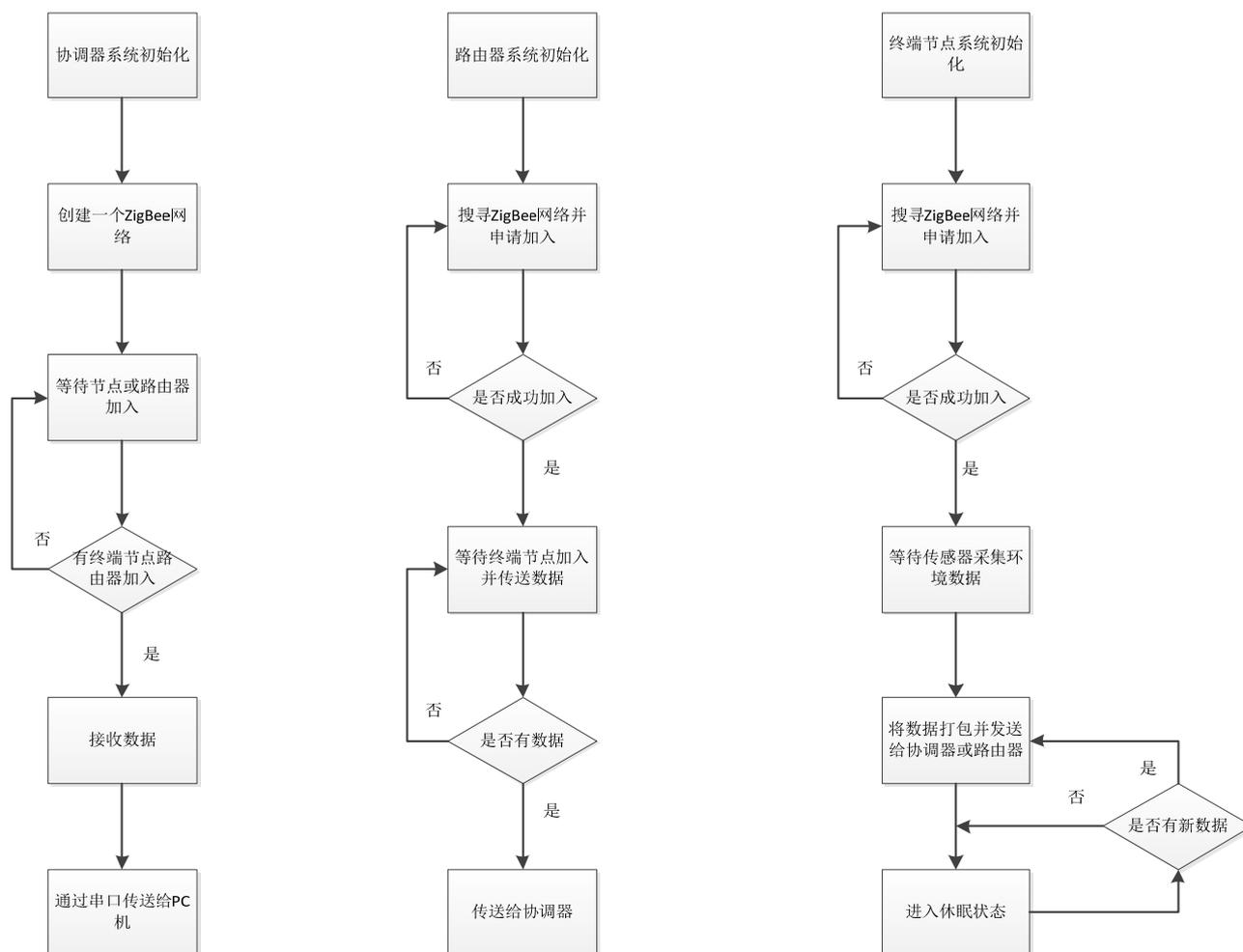


Figure 6. Boot workflow of zigbee  
图 6. ZigBee 启动流程图

**NWK:** 网络层目录, 含有 NWK 层参数的配置文件、NWK 层库的函数接口文件和 APS 层库的函数接口。

**OSAL:** 协议栈的操作系统抽象层, 可以实现多任务为核心的资源管理机制。**Z-Stack** 是通过类似 uCOS 操作系统实现的, 协议的各层都被分配了一个任务, 可以按照一定的规则进行管理。

**Profile:** AF 层目录, 含有 AF 层处理函数的接口文件。

**Security:** 安全层目录, 含有该层的处理函数接口文件。

**Services:** 地址的处理函数目录, 含有地址模式的定义和地址的处理函数。

**Tools:** 工程配置目录, 含有 Z-Stack 的一些相关配置信息。

**ZDO:** ZDO 目录, 一种公共的功能集, 使用户很方便地用自定义的对象调用 NWK 层和 APS 子层的服务。

**ZMac:** ZMac 目录, 含有 Z-Stack MAC 导出层的接口文件和一些 NWK 层的函数。

**ZMain:** 主函数目录, 含有入口函数和对硬件配置的接口函数。

**Output:** 输出文件目录, 由 IAR EW8051 IDE 自动产生。

底层协议是在 TI 公司提供的 Z-Stack 协议基础上, 结合实际硬件电路进行设计。这部分用 C 语言编写, 分为协调器、路由器和采集节点三种类型。本文以 TI 公司提供的 SampleApp 例子为基础, 在协议栈 APP 目录输入下加入 DS18B20.C 和 DS18B20 的源文件; 接着在 APP 找到 OS\_SampleApp.c 文件下的 tasksArr 数据, 可以看到会执行的任务; 只需要用它提供的 API, 在 SampleApp\_ProcessEvent 函数, 中有各种对应的事件处理函数。在 SAMPLEAPP\_SEND\_PERIODIC\_MSG\_EVT 事件中加入温度采集函数, 并将通过传感器采集的数据通过 SampleApp\_SendPoint-Message 函数将接受到的数据发生出去。数据发送出去后, 通过路由器转发, 最后传到协调器。在 Sample App\_ProcessEvent 函数中, 当有数据接受事件发生时, 就会进入 AF\_INCOMING\_MSG\_CMD 事件, 在 Sample App\_MessageMSGCB 函数中, 通过 afIncoming MSG Packet\_t 类型的指针, 找到接受到的数据存放位置, 并通过 HalUARTWrite 函数通过串口传输给 PC 机<sup>[6]</sup>。

在服务器数据处理部分, 本文采用的是 Linux 系统下 web 服务器。利用 open() 函数打开 /dev/ttyS1 文件, 并按照串口传输协议规范提取出数据信息。并通过 Linux 下的 time 等相关函数获取本地时间, 用 open() 函数创建另外一个文件用来存放此时的时间和串口传送过来的有用信息。然后建立一个 web 服务器, 通过 socket() 函数实现服务器的网络通信, 并设置 sockaddr\_in 结构体中的各项参数, 给 bind() 函数和 listen() 函数传入参数, 使服务器运行起来, 并用 fork() 函数创建多进程, 最后通过 accept() 函数来等来用户访问。当有用户访问时, 读取文件中有用信息, 并通过 write() 函数将数据发送给用户。如有需要可以使用 Linux 下 MySQL 数据库对数据进行增删改查<sup>[7]</sup>。如图 7 是服务器启动流程图。

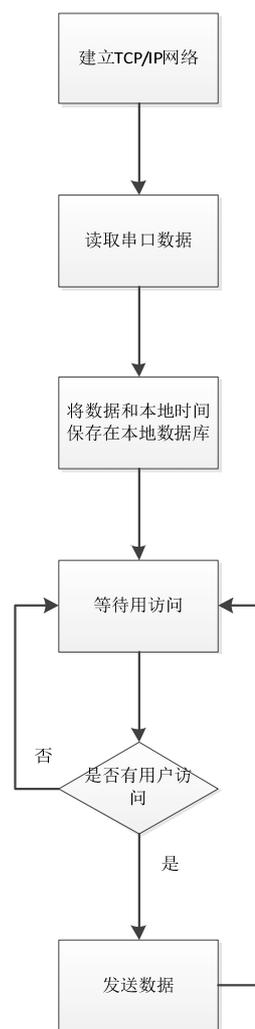


Figure 7. Workflow of server  
图 7. 服务器流程框图

在智能终端客户端(如智能手机)与服务器通信的流程如下: 如图 8 所示。

- 1) 通过给定的 url 地址生成一个 URL 对象;
- 2) 通过 URL 对象的 `openConnection()`方法得到一个 `URLConnection` 对象 `con`;
- 3) 用 `con` 的 `setRequestMethod(String method)` 方法设置 URL 的请求方法如: `GET`、`POST` 等,如果要向服务器发送一些消息头属性,可以用 `con` 的 `setRequestProperty(String key, String value)`方法来设置请性属性如: `setRequestProperty("Referer","");`
- 4) 通过 `con` 的 `getOutputStream()`方法得到于服务器联通的输出流对象 `outputStream`, 这样就可以通过 `outputStream` 的 `write` 方法向服务器写请求参数数据
- 5) 然后 `con` 的 `getInputStream()`方法得到与服务器联通的输入流对象 `inputStream`, 这样就可以通过 `inputStream` 的 `read` 方法获取从服务器那返回的数据。
- 6) 获取服务器的数据后, 跟据情况通过 `outputStream` 对象的 `write` 方法向服务器写相应的命令数据。

## 5. 系统调试与数据采集

环境数据采集前, 通过 IAR Embedded 的开发环境对程序进行编译下载, 根据硬件电路需求下载到相应的协调器、路由器和终端节点。下载完成后, 根据环境的实际情况进行相应布点, 建立一个运行正常的网络通道。

在协调器启动后, 等待路由器或终端节点的加入。路由器启动后等待终端节点加入。终端节点上电复位后, 首先搜寻所在区域的网络, 并提出加入网络的申请, 协调器允许后便成功加入网络, 并从串口输出数据信息。如图 9 所示。

## 6. 总结

本文设计的基于 ZigBee 技术的网络环境监测系统, 结合 ZigBee 无线通信、串口通信、Linux 下网络编程和安卓系统开放等技术, 成功完成了对指定环境因素的数据采集, 搭建出环境监测的物联网系统。

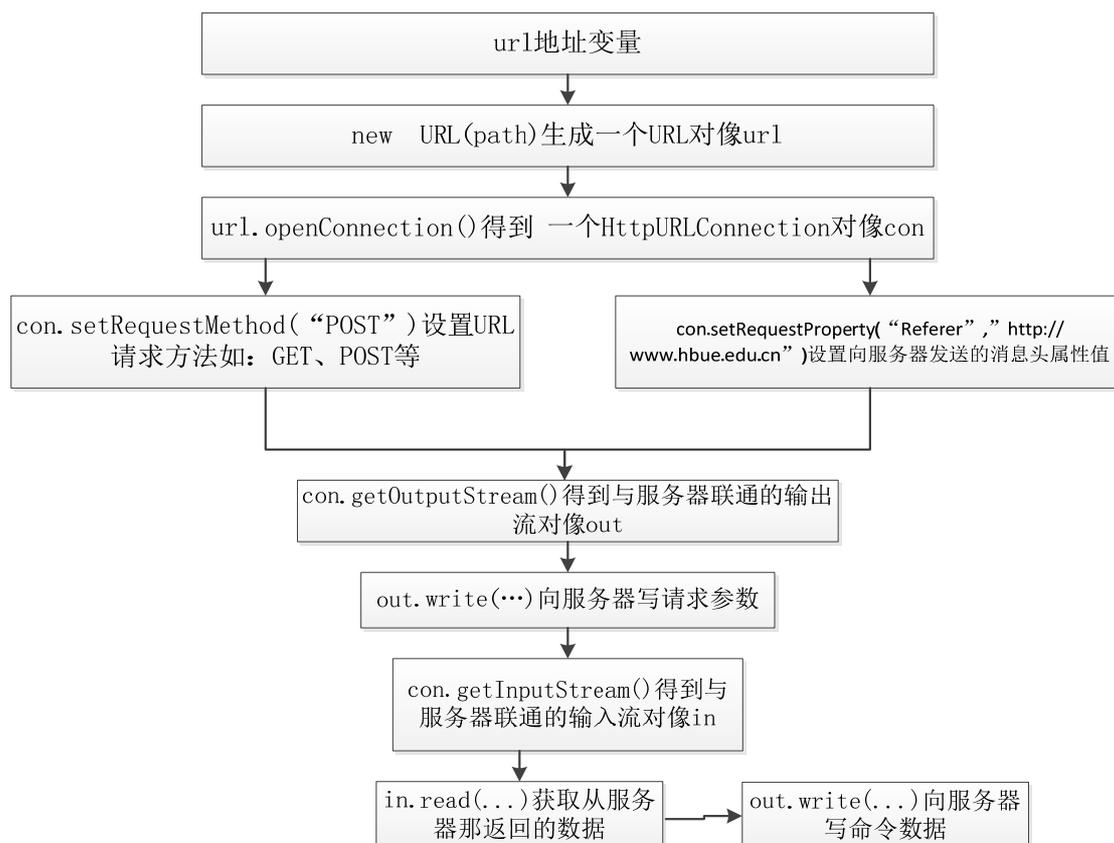


Figure 8. Workflow of communications  
图 8. 与服务器通信流程图



Figure 9. Temperature collection  
图 9. 采集温度

## 参考文献 (References)

- [1] 高守玮, 吴灿阳, 杨超等. ZigBee 技术实战教程——基于 CC2430/31 的无线传感网络解决方案[M]. 北京: 北京航空航天大学出版社, 2009, 3-10.
- [2] 刘伟荣, 何云. 物联网与无线传感器网络[M]. 北京: 电子工业出版社, 2013.
- [3] 王小强, 欧阳俊, 黄宁淋. ZigBee 无线传感网络设计与实现[M]. 北京: 化学工业出版社, 2012.
- [4] 翟雷, 刘盛德, 胡咸斌. ZigBee 技术与应用[M]. 北京: 北京航空航天大学出版社, 2007, 3-19.
- [5] 吕志安. ZigBee 原理与应用开放[M]. 北京: 北京航空航天大学出版社, 2007, 3-19.
- [6] 杨丰盛. Android 应用开发揭秘[M]. 北京: 北京机械工业出版社, 2010.
- [7] 余志龙, 陈昱勋, 郑名杰. Android SDK 开发范例大全[M]. 北京: 北京人民邮电出版社, 2010.