基于 ZigBee 的物联网环境数据采集与控制系统①

张美平, 许 力

(福建师范大学 福建省网络安全与密码技术重点实验室, 福州 350007)

摘 要: 针对目前大部分无线传感器网络应用系统设计方案仅实现了数据采集功能、而没提供控制功能的缺点, 结合物联网环境数据采集与控制系统的需求, 提出了一种基于 ZigBee 无线通信技术的物联网数据采集与控制系统的设计方案. 应用内置 EmberZNet 协议栈的无线 ZigBee 单片机 STM32W108 设计了物联网节点, 开发了相关的硬件节点软件代码与管理软件, 实现环境数据的采集与控制功能.

关键词: ZigBee; STM32W; 物联网; 环境数据采集与控制

IOT Environmental Data Acquisition and Control System Based on ZigBee

ZHANG Mei-Ping, XU Li

(Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350007, China)

Abstract: Most of the wireless sensor network application systems only realize the function of data acquisition and don't provide control function. For the need of environment data acquisition and control system, an IOT environmental data acquisition and control system based on ZigBee was proposed in this paper. The hardware is designed by STM32W108 which with built-in EmberZNet ZigBee protocol software was developed to realize the remote environment data acquisition and control system.

Key words: ZigBee; STM32W; internet of things; data acquisition and control

环境参数采集与控制系统是物联网中一种典型应用, 在精细农业种植如大棚监测与控制、野外环境监测应用如森林碳排放量检测、以及水资源环境检测等应用领域有着重要的意义. 环境参数采集与控制系统通常采用无线传感器网络作为其底层核心的通信技术, ZigBee 是无线传感器网络(wireless sensor network, WSN)中的一种典型的通信协议, 以其低功耗、低成本、组网自适应、网络容量大等特点, 被广泛应用于自动控制和监控领域[1,2,6].

研究人员提出了多种建立 ZigBee 应用系统的设计方案, 文献[3]提出了一种利用 CC2430 芯片设计煤矿井下环境监测系统, 文献[4]提出了一种利用无线传感网络设计葡萄园温室监控系统. 大部分基于 ZigBee 无线传感器网络的应用系统仅实现了数据采集功能, 即分布在不同位置的无线传感器节点把采集到的数据发送给 Sink 汇聚节点, 汇聚节点接收到数据后传输到指定的服务器. 作为数据采集与控制类的物联网应用系统, 需要在采集节点执行数据采集功能的同时还要具备向部分物联网节点发送控制指令的功能, 因此在实现数据采集功能的基础上进一步实现控制功能就显得尤其重要. 本文提出了一种利用 STM32 微处理器与 STM32W108 ZigBee 单片机实现物联网数据采集与控制功能的方案, 利用 EmberZNet、EmberZNet-EZSP 协议栈编写物联网数据采集与控制节点的软件代码, 并开发后台管理软件, 实现了一种物联网环境数据采集与控制系统.

1 环境数据采集与控制系统硬件设计

1.1 物联网环境数据采集与控制系统总体框架

当前具有无线通信功能物联网节点通常采用基于蓝牙、Wifi、无线 ZigBee 三种技术设计节点. 应用蓝牙技术设计的节点可以组成星型网络, 其组成的网络

① 基金项目:福建省教育厅科技项目(JK2011010);福建省自然科学基金(2013J01222)
收稿时间:2014-03-27;收到修改稿时间:2014-05-19

具有覆盖范围小的缺点; 基于 Wifi 技术设计的节点需要接入到固定的 WLAN 网络, 具有网络覆盖范围大、可用带宽大、功耗大的特点, 适合用于设计视频监控类的物联网应用, 但依赖于 WLAN 基础网络设施; 基于无线 ZigBee 单片机的设计的节点具有功耗低、支持 Mesh 组网, 多个 ZigBee 节点利用无线 Mesh 多跳组网组成一个通信范围更大的网络, 适合建立需要大规模部署节点的远程环境参数采集与控制的物联网系统.

在 ZigBee 无线通信系统中, 有协调器节点和终端节点、路由节点三种角色. 协调器节点负责建立 ZigBee 网络、接收终端节点采集的数据, 路由节点负责实现多跳 Mesh 组网、扩展网络的覆盖范围、提供数据中继转发功能, 终端节点采集相关环境数据, 通过路由节点发送给协调器或 Sink 汇聚节点.

本文提出的物联网环境数据采集与控制系统采用 ZigBee 作为其无线通信技术, 系统由协调器、路由节点、环境数据采集与控制节点、以及环境数据采集与控制系统服务器组成, 系统组成的结构如图 1 所示. 整个网络有一个 ZigBee 协调器以及若干 ZigBee 路由节点以及终端节点、控制节点组成. 环境数据采集与控制系统服务器与 ZigBee 协调器通过 USB 接口连接, 运行相应的数据采集与控制管理软件; 各个数据采集节点采集相应的环境参数发送个 ZigBee 协调器, 协调器把从终端节点接收的数据通过 USB 接口发送给环境数据采集与控制管理主机, 管理软件处理接收到的数据包, 同时管理软件也可以对控制节点发出对应的控制指令.

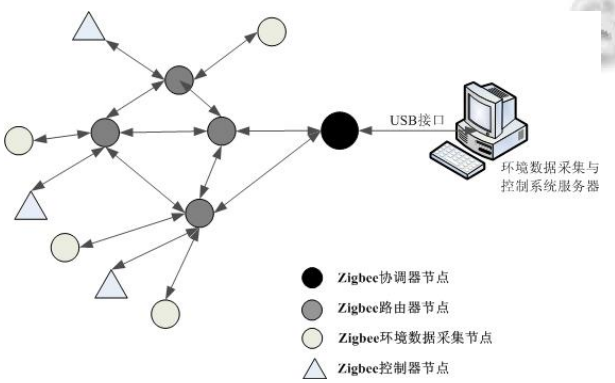

图 1　系统总体框架图

1.2 物联网节点硬件设计方案

采用 ZigBee 单片机设计物联网节点的硬件, 通常有以下几种模式: SOC(System on Chip) 模式、MCU+NCP(Network Coprocessor)模式、MCU+ZigBee 透传模式[7].

STM32W108 是 ST 意法半导体公司推出的系统级 ZigBee 芯片[5], 其无线通信部分符合 IEEE 802.15.4 的标准, STM32W108 支持 EmberZnet ZigBee 协议栈, 能以单芯片 SOC 模式设计物联网节点, 直接调用 EmberZnet 提供的 ZigBee 协议栈 API 实现 ZigBee 组网通信的功能; 同时 STM32W108 也支持 EmberZnet_EZSP 串行通信协议, 可以采用 MCU+NCP(Network Co-Processor)模式设计物联网节点, 把 STM32W108 芯片通过 SPI 接口与 STM32 MCU 微控制器连接, 在对应的 MCU 的中调用 EmberZnet_EZSP 串行通信协议与 STM32W108 芯片通信, 实现 ZigBee 组网通信的功能.

本系统采用 MCU+NCP 模式来设计环境数据采集与控制系统的 ZigBee 协调器, 核心微控制器采用 32 位 ARM Cortex-M3 为内核的微控制器 STM32F103, 使用支持 EmberZnet EZSP 协议的 STM32W108 作为 ZigBee 网络通讯模块, 其高效 ZigBee 网络协议栈 EmberzNet 完成了对所有 ZigBee 网络功能的封装, 微控制器与 ZigBee 通讯模块之间采用 SPI 接口连接, 在 STM32F103 微控制器上调用 EmberZnet_EZSP 串行协议即可实现无线 ZigBee 组网与通讯. STM32F103 还提供了大量的外部 GPIO, 可以实协调器的 LED、按键、蜂鸣器使其具有丰富的交互功能; STM32F103 芯片还自带一个 USB 接口, 并提供 USB 虚拟串口驱动, 可以实现通过 USB 接口与 PC 上位机的通信, 利用该方式设计的硬件可以轻松实现 ZigBee 协调器的功能, 硬件结构如图 2 所示.

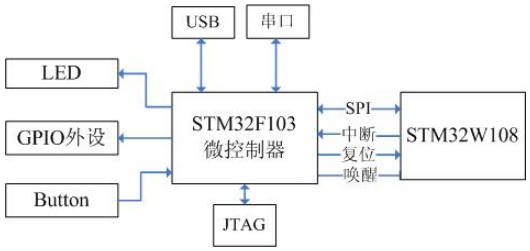

图 2　ZigBee 协调器硬件设计框图

环境数据采集与控制节点以及 ZigBee 路由节点采用 SOC 单芯片设计方案, 使用 STM32W108 作为节

点核心处理器. STM32W108 与目前其他的 2.4GHz SOC 芯片最大的区别在于: 低功耗、采用了 32 位 ARMCortex-M3 内核、处理能力强. 芯片内部带有功率放大器, 可获得较大的通信距离. STM32W108 微控制器提供了丰富的 GIPO 接口, 可以当作连接外部设备的控制接口, 如用于提供 LED、按键、蜂鸣器接口, 使节点具有丰富的人机交互与设备控制功能; 同时 STM32W108 提供丰富外设接口如 Uart、ADC、I2C、SPI 的接口, 可以方便外接多种传感器芯片, 如通过 I2C 接口外接 TSL2561 光强传感器连接 SHT11 温湿度传感器, 实现对环境的光照强度、温湿度等环境数据的采集. 其硬件结构如图 3 所示.

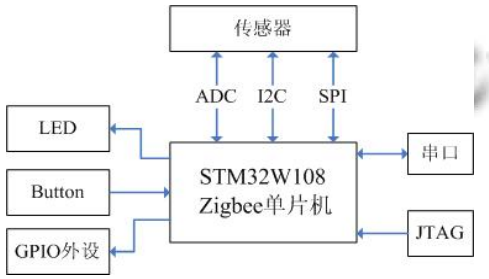

图 3 ZigBee 路由、终端、控制节点硬件设计框图

所设计的硬件节点可以同时作为数据采集与控制以及路由节点使用. 数据采集节点外接传感器, 控制器节点通过外接对应的控制外设, 在软件设计时把 ZigBee 协议栈的工作模式设置为终端模式; 路由节点, 不外接控制模块与传感器模块, 编程时设置其协议栈的工作模式为路由模式, 为系统中的数据采集与控制节点实现 ZigBee Mesh 多跳组网与接入功能.

2 物联网环境数据采集与控制系统软件设计

2.1 EmberZnet 协议栈

STM32W108 芯片配套的软件开发工具是由 IAR 公司提供的专门针对 STM32W108 推出的 EWARM-5.41.2 集成开发环境. EmberZNet 协议栈是 ST 公司提供开发工具包, 该工具包提供了智能能源/家庭自动化(SE/HA)、传感器数据采集(sensor/sink)的等多种应用的 Demo 例程, 用户可以在例子 Demo 代码的基础上编写 STM32W108 芯片上的应用代码; 同时 ST 也提供了 EmberZNet_EZSP 的协议接口函数及其 Demo 代码, 以方便 MCU+NCP 模式设计的节点编写使用对应的代码[5].

EmberZNet 协议栈内部包含了丰富的 API 函数, 用户只需调用对应的 API 可完成网络组建、数据接收, 数据发送等功能代码的编写; 如在协调器端调用 emberFormNetwork()可以组建一个 ZigBee 网络, 在节点端调用 emberJoinNetwork()可以使节点加入到某一个 ZigBee 网络, 调用 emberSendUnicast()函数可以向指定的节点发送单播数据; 用户需要根据实际需求编写协议栈事件对应回调函数, 以实现同 EmberZnet 协议栈的交互, 如当协议栈接收到信息时, EmberZNet 协议栈会触发对应的数据接收事件, 对接收到数据的处理代码就要在 emberIncomingMessageHandler()的函数中实现.

协调器节点采用 STM32+STM32W108 的组合设计硬件, 协调器的应用软件的开发过程中需要在对应的 STM32 编程过程中调用 EmberZNet_EZSP 协议来控制 ZigBee 网络处理器上的 EmberZNet 协议, 其相关的接口 API 功能基本一致, 只是大部分的 API 函数名称改成了 ezsp 开头, 如向节点发送单播数据包的 API 函数为 ezspSendUnicast().

2.2 应用 EmberZNet 协议栈设计节点软件代码

EmberZNet 协议栈整体结构类似于一个微控制器操作系统, 应用 EmberZNet 协议栈设计的节点主程序的结构图 4 所示.

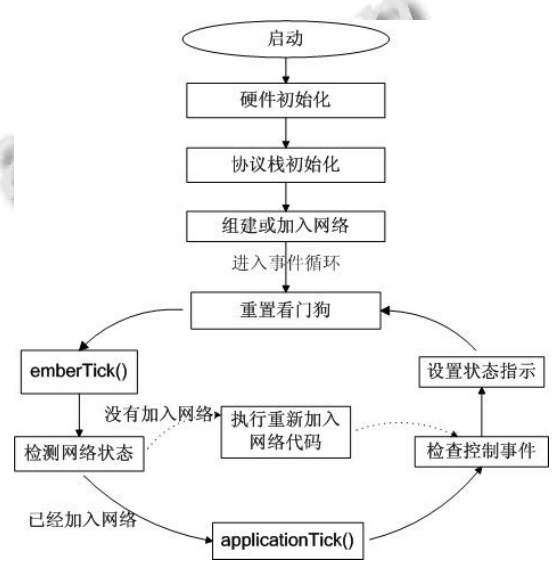

图 4 基于 EmberZnet 的应用主程序结构图

程序启动后, 执行硬件初始化指令、emberinit()、设置相关的 ZigBee 网络参数、执行加入网络的指令, 然后进入到整个协议栈的事件的主循环中, 分别执行

embertick(), 然后根据节点入网的状态执行applicationtick()或执行重新加入网络的代码.

数据采集终端与控制节点需要在协议栈初试化完成后执行加入网络的代码, 同时在 applicationtick()中编写传感器数据采集与发送的代码, 在emberIncomingMessageHandler()回调函数中编写对接收到 ZigBee 数据包后的处理代码.

协调器节点在协议栈初试化完成后执行组建网络的代码, 并设置允许其他节点加入网络, 编写ezspIncomingMessageHandler()回调函数的代码实现把接收到的传感器数据的转发到 usb 虚拟串口, 在applicationtick()中编写对应的代码把环境数据采集与控制系统服务器管理软件通过 USB 虚拟串口发送的控制指令转发给特定控制节点.

2.3 数据采集与控制系统节点地址

ZigBee Pro 协议栈中有两种形式的节点地址, 一种是 64 位 IEEE EUI-64 MAC 地址, 一种是 16 位的ZigBee 短地址,这两种地址均可以用于 ZigBee 节点间的通信. 64 位的 EUI-64 地址是固定在 ZigBee 芯片中的物理地址, 类似以太网络中的网卡物理地址, 每个ZigBee 芯片具有唯一的 EUI-64 地址. EmberZNet 协议是一种 ZigBee Pro 标准的协议栈, 多个节点围绕一个中心协调器, 支持 Mesh 多跳组网从而扩展网络覆盖范围, 并使用一个随机的 16 位地址寻址模式来形成网络. 16 位的 ZigBee 短地址是在 ZigBee 网络启动后由网络协议栈动态临时分配的, 节点每次上电启动后加入网络, 或在运行过程中执行重新入网的指令加入网络后, 会分配一个可能与前一次入网时所分配的不同的16 位 ZigBee 短地址. 图 5 为使用 IEEE802.15.4 无线协议分析仪抓取了一个 EUI-64 的地址为 00:80:e1:02:00:1b:e2:be 的节点, 第一次接入网络后获得的 ZigBee 地址为 0x2fa2, 再次接入网络后 ZigBee 地址获得的 ZigBee 短地址为 0xc5ce.

由于数据采集与控制节点每次入网获得的 ZigBee 地址可能发生变化, 采用 ZigBee 地址将不能在系统中唯一识别某一个物联网节点, 因此不能采用节点的ZigBee 地址作为控制系统对节点的唯一编号, 在前期基础工作[7]中提出了一种给每个节点分配一个固定的节点编号, 数据采集与控制节点周期性地向协调器发送包含有“节点编号、ZigBee 地址、EUI-64 地址”的心跳注册包, 并在协调器中建立节点编号、ZigBee 地址、EUI-64 地址与的映射表的方法来解决这个问题. ZigBee 协调器在 usb 虚拟串口端接收到管理端发来的给指定节点的控制指令时, 查询地址映射表找出当前节点的最新 ZigBee 地址, 把控制指令通过 ZigBee 网络发送给指定的控制节点.

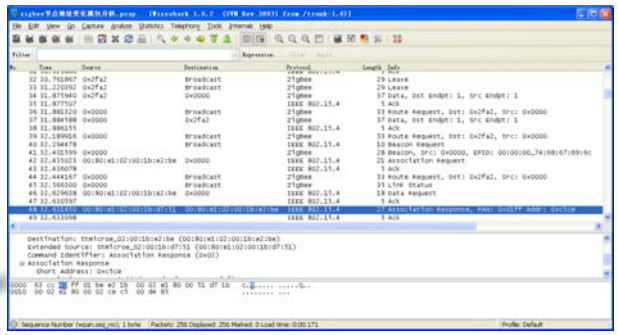

图 5　节点 2 次接入 ZigBee 网络后的短地址变化的抓包分析

2.4 物联网节点间的通信过程

在建立 ZigBee 网络时, 要先启动协调器节点, 协调器根据预先设置的网络 PAN ID、无线网络频段等信息负责建立 ZigBee 网络, 组网成功后分配给协调器的16 位 ZigBee 短地址为 0x0000; 路由节点与终端设备启动后, 自动寻找 ZigBee 网络并加入到组建好的ZigBee 网络, 加入成功后节点获得由协调器动态分配16 位的 ZigBee 短地址. ZigBee 节点间的单播通信就使用目标节点的 16 位 ZigBee 短地址作为其发送 API 的目标地址, 在 EmberZNet 协议栈编程过程中用户可以调用 emberSendUnicast()、ezspSendUnicast()来实现向特定的节点发送数据的功能.

在传感数据采集与上报阶段, 传感器节点把采集到的数据按一定的格式打包后把数据发送给 ZigBee地址为 0x0000 的协调器、协调器再把数据通过 USB 接口转发给 PC 管理软件.

在 PC 管理软件发送控制指令给传感器节点的阶段, PC 端的管理软件把要发送给特定节点的指令按控制指令格式打包, 写入 USB 虚拟串口, 在协调器端编写对应的 USB 虚拟串口数据接收代码, 接收到控制指令数据包后, 取出控制指令的目标节点地址, 查找节点映射表, 找出对应的 ZigBee 地址, 把控制指令转发给对应的节点, 实现 PC 管理软件对传感节点的控制功能.

协调器端的处理来自传感器节点的数据“ezspIncomingMessageHandler()”的部分代码如下:

```
switch (apsFrame->clusterId) {
    case REGISTION:
        for(i=0;i<8;i++)
        {
            theEui64[i]=message[i];
        }
        enpnum=(message[8]<<8)+message[9];
    nodeidindex = Append(theEui64, sender,enpnum);
    returnIndex[8] = nodeidindex & 0xff;
    returnIndex[9] = (nodeidindex >> 8) & 0xff;
    MEMCOPY(&(returnIndex[0]), emberGetEui64(), EUI64_SIZE);
    sendUnicast(sender , REGISTION , returnIndex , 2+EUI64_SIZE);
    break;
case SENSOR_REPORT:
    MEMCOPY(sendBuffer, message, length);
    usbTXFinish = 1;
    UserToPMABufferCopy(sendBuffer, ENDP1_TXADDR, length);
    SetEPTxCount(ENDP1, length);
    SetEPTxValid(ENDP1);
while(usbTXFinish == 1);//wait while usb is transmitting
        break;
  }
```

这里协调器接收到传感器节点发来的数据包后, 根据 apsFrame->clusterId 来识别是传感数据还是注册数据, 如果是“SENSOR_REPORT”传感数据, 则直接调用 USB 的发送代码, 把数据发送给 PC 管理软件; 如果是“REGISTION”节点注册数据, 则提节点编号 enpnum, 节点的 EUI-64 地址、节点的 Zigbee 地址, 并调用 Append()函数, 向链表注册该节点的信息.

协调器处理来自 PC 管理软件的控制指令的代码如下:

```
nodeidindex=PC_cmd[commandNum-1][4]+(PC_cmd[commandNum-1][3]<<8);
    sendUnicast(FindShortIdByEnpNum(nodeidindex), PC_cmd[2], PC_cmd,8);
```

这里 PC_cmd 为从 usb 接收到的控制指令数组, 提取需要控制的节点编号保存在 nodeidindex, 然后调用 FindShortIdByEnpNum()函数从链表中查找该节点对应的 ZigBee 地址, 并调用 sendUnicast()向节点发送对应控制指令.

3 系统运行测试

在完成相关硬件设计后, 组建了一个以 5 个温湿度传感器节点、3 个光照传感器节点的物联网环境参数采集与控制的测试系统, 其硬件节点与系统实物如图 6 所示.

(a) stm32w节点硬件 (b) 系统硬件连接实物图

图 6 物联网环境参数采集与控制硬件系统连接示意图

编写相关的物联网环境控制与控制的节点软件代码与 PC 端管理软件, 并在 PC 主机端安装好 STM32 CDC-USB 虚拟串口驱动, 运行对应的上位机程序. 节点代码在发送数据时, 把当前节点号、到协调器路由的下一跳地址、以及节点的 EUI64 MAC 地址、传感器采集到的数据数据、以及 2 个代表继电器运行情况的 GPIO 的电位状态也一并发送到 PC 上位机的数据处理软件. 图 7 是使用 wireshark 解码 IEEE 802.15.4 的数据帧, 编号为 4004 的温湿度节点发送给协调器的数据包解码包, 节点 4004 向协调器发送“7e:42:40:40:04:c5:ce:be:e2:1b:00:02:e1:80:00:00:00:00:22:00:36:00:00:00:00:00:00:7e”, 数据包表示节点编号为 4004, 数据包的接收目的地址 0x0000, 节点的 16 位短地址为 0xc5c3, 节点的 EUI-64 地址00:80:e1:02:00:1b:e2:be, 温度数据为0x22(34度), 湿度数据为 0x36(54).

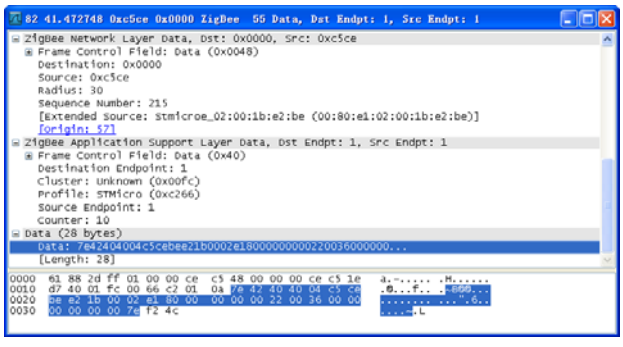

图 7 wireshark 抓取 4004 节点发送数据包解码

把 8 个节点分别放在实验楼的 4、5 楼, 测试结果显示, 多个节点通过 Mesh 功能成功组网, 控制指令能快速发送到指定的节点. 从图 8 的网络运行网络拓扑图看到 8 个节点组成一个多跳通信的网络, 扩展了网络的覆盖范围, 图 9 显示了节点 4002 当前采集到的相关数据, 同时也可以通过 LD1 OFF 以及 LD2 OFF 控制按钮, 发送特定的控制指令, 实现对某一个特定节点的的 GPIO 外设的控制.

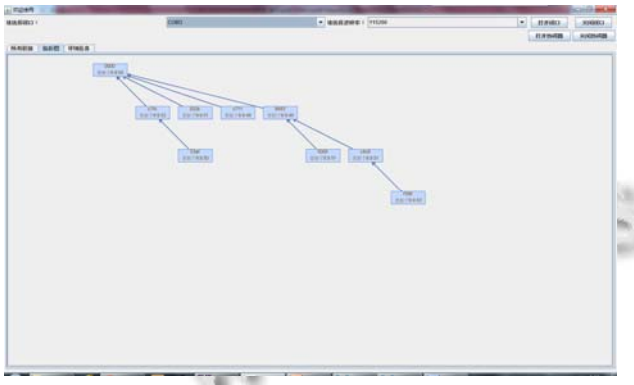

图 8　PC 管理软件中展示的各个节点到协调器的路由拓扑图

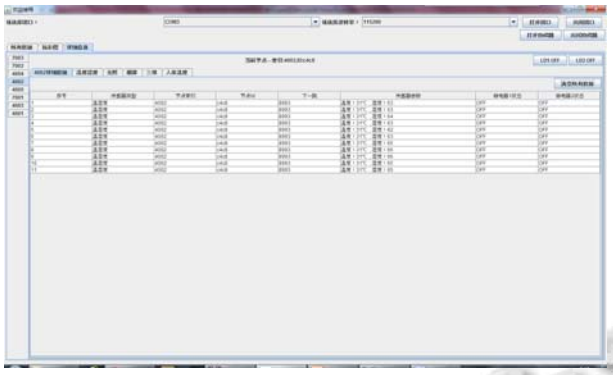

图 9　节点 4002 的具体环境参数数据

实验测试表明, 本系统实现的环境数据采集与控制系统能正常运行.

4　总结与展望

实验测试过程中发现一个 ZigBee 地址更新不同步而影响控制指令成功发送的问题, 有待于进一步解决, 即当数据采集和控制节点由于某些原因使得某个节点当前 ZigBee 地址刚好发生改变, 而协调器节点中的注册信息由于未收到对应的新 ZigBee 地址的注册信息导致地址映射表没有及时更新, 此时管理控制软件如果刚好发出的控制指令将不能顺利地由协调器发送给数据采集与控制节点, 导致控制指令丢失, 影响系统执行控制指令的有效性. 对于这个问题将有待于进一步的解决. 这个也后续的研究工作要进一步解决的问题.

对于环境数据采集与控制类的的物联网控制系统, 网络规模相对较大, 所需要的终端数目较多, 所以对终端节点的硬件成本较敏感, 文献[7]提出的方案中所有的 ZigBee 节点均采用 MCU+NCP 的模式设计物联网节点, 这种方式的硬件成本较高, 本文提出的数据采集与控制系统设计方案中仅协调器节点由于需要处理更多任务与通信任务而采用 MCU+NCP 方案设计协调器, 系统中需要大量安装部署的数据采集与控制终端与路由节点采用支持 ZigBee SOC 单芯片的设计方案, Stm32w108 的单芯片方案的硬件成本比 Stm32+SN260 的 MCU+NCP 的方案的成本低很多, 因此本系统使用的方案在硬件成本上有较大的优势.

本文提出的采用 STM32F103 微控制器与 STM32W108 ZigBee SOC 芯片相结合的物联网环境数据采集与控制系统的设计方案可以用于多种需要无线数据采集与控制的物联网应用场合, 具有一定的实际应用意义与推广价值.

参考文献

1 ZigBee Alliance. Network specification(draft version 1.0). http://www.zigbee.org/.[2012-02-12].

2 孙利民,李建中,陈渝.无线传感器网络.北京:清华大学出版社,2005.

3 刘杰,邓志东,杨鹏,董志然,裴忠民.基于 ZigBee 协议的煤矿井下嵌入式系统.计算机应用,2008,28(12):302–303.

4 陈金凯,曹剑炜,陈庆奎.ISHSN:一种异构传感网融合系统.计算机应用,2013,33(5):1191–1193.

5 沈建华,郝立平.STM32w 无线射频 ZigBee 单片机原理与应用.北京:北京航空航天大学出版社,2010.

6 刘云浩.物联网导论.北京:科学出版社,2010.

7 张美平.基于 STM32 与 SN260 的物联网数据采集与控制系统.计算机系统应用,2013,22(11):86–89.