

HTTP 协议与 JSON 格式在 ZStack 中的实现与应用

卢少军 过丹婷 刘守印

(华中师范大学 物理科学与技术学院 武汉 430079)

摘要: ZStack 是 TI 公司推出的符合 ZigBee 标准的协议栈,专注于构建低功耗、低速率的无线传感器网络。针对该协议栈无法直接接入互联网,提出了 HTTP 协议在 ZStack 中的应用方案。首先通过以太网芯片 ENC28J60 将 uIP 协议栈移植到 ZStack 中,实现了 ZigBee 网络便捷接入互联网;其次对 uIP 协议栈中的 TCP 客户端模式进行封装,实现了 HTTP 协议的 POST 请求,采用互联网中通用的 JSON 标记语言传输数据到 PC 数据库,保证了通信的简单高效;最后采用此方案采集温湿度信息并传输到后台,在 WEB 端实现温湿度曲线的实时绘制与 ZigBee 网络的拓扑显示。

关键词: ZStack;uIP;HTTP 协议;JSON;实时监测

中图分类号: TP274.2 **文献标识码:** A **国家标准学科分类代码:** 510.1020

Implementation and application of HTTP protocol and JSON format in ZStack

Lu Shaojun Guo Danting Liu Shouyin

(College of Physical Science and Technology, Central China Normal University, Wuhan 430079, China)

Abstract: TI ZStack, a protocol stack up to ZigBee standard, focused on building low power, low rate wireless sensor network (WSN). Aiming at it's not direct access to the Internet, the application of HTTP protocol in ZStack is provided. Firstly, With the aid of the network chip ENC28J60, uIP protocol is ported to the ZStack providing the ZigBee easy access to the internet. Secondly, encapsulating the TCP client mode in the uIP protocol stack realizes the POST requirement in the HTTP protocol. JSON markup language widely used in the internet is adopted to transfer data to the PC database so the simple and effective communication is ensured. Finally, with the use of this program collecting temperature and humidity and transfer the related data to backstage, and realize the real-time data curve drawing of temperature and humidity and displaying of ZigBee network topology.

Keywords: ZStack;uIP;HTTP protocol;JSON;real-time monitoring

1 引言

ZStack 作为实现 ZigBee 标准的半开源协议栈,由其构建的网络具有自组网、低功耗、频段免费、成本低等特点,因而被广泛应用在环境监测、工业控制、医疗监控等领域进行前端数据采集^[1]。随着信息化和物联网的概念深入人心,人们迫切希望将各种设备接入互联网,来享受诸多便捷服务。

WEB 应用是互联网蓬勃发展的产物,社交网络、电子商务等极大地改变了我们的生活方式。HTTP 协议^[2]作为 WEB 数据传输的主流标准,使用起来简单快速、稳定高效,俨然已成为互联网最重要的应用层协议之一。JSON^[3-4]是一种通用、安全的数据格式,在跨平台数据传输和数据解析中具有无可比拟的优势。

本文将 HTTP 协议应用到 ZStack 中,并通过 JSON 格式传输数据,结合三者优势实现 ZigBee 网络无缝接入互联网。首先介绍了 uIP 协议栈^[5]移植到 ZStack 中的方法,并对 uIP 协议栈的 TCP 客户端方式进行 HTTP 封装,实现了 CC2530 直接作为 ZigBee 以太网网关^[6]的需求;在数据传输上采用 JSON 打包,格式统一简单,降低了后台解析时的复杂度。最后通过对温湿度的实时监测,验证了数据通过 HTTP 协议在 ZigBee 网络和以太网间的可靠传输。

2 ZStack 与 uIP 协议栈

ZStack 是 TI 公司推出的一款业界领先的 ZigBee 协议栈,提供了大量供开发人员使用的应用程序接口(API)与服务(services),用户可根据需求组建星型(star)、树形

(tree)和网状(Mesh)拓扑的无线传感器网络。

ZStack 架构如图 1 所示,包括应用层框架(AF)、应用支持子层(APS)、网络层(NWK)、安全服务提供层(SSP)以及 IEEE802.15.4 规定的物理层(PHY)和媒体接入控制层(MAC)。

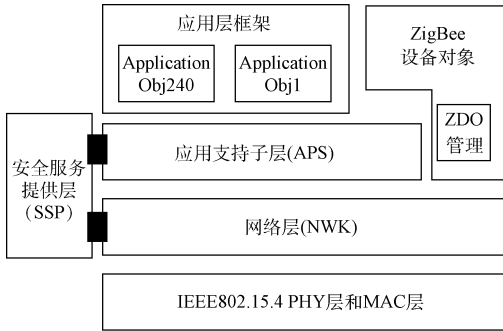


图 1 ZStack 协议栈架构

协议栈采用分层结构设计,使各层相对独立,每一层向上一层提供一些服务的同时可以向下一层发出请求,这样开发人员只需要关注与自己应用相关的那部分协议^[7-8]。

整个 ZStack 依靠操作系统抽象层(OSAL)维护任务(task)队列与事件(event)队列,通过时间片轮转实现任务调度。协议栈首先通过 osalInitTasks()实现各层任务初始化,使用 TaskID 作为每个任务的唯一标识号;接着调用 osal-run-system() 函数进入死循环。在系统心跳的驱动下,OSAL 不断通过 do-while 按优先级轮循事件队列 tasksEvents[],一旦有事件发生,则通过任务队列 tasksArr 找出相应任务处理函数进行处理,处理完成之后继续轮循下一事件。如果任务列表中没有就绪任务,则处理器进入休眠状态实现低功耗。其运行流程如图 2 所示。

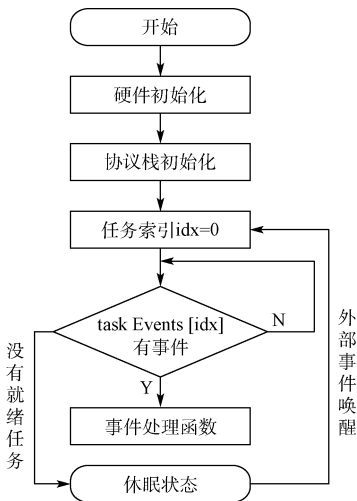


图 2 OSAL 执行流程

uIP 是由瑞典计算机科学学院网络嵌入式系统小组的 Adam Dunkels 专门为嵌入式系统设计和开发的免费开源

TCP/IP 协议栈。它去掉了完整的 TCP/IP 中不常用的功能,简化了通讯流程,但保留了网络通信必须使用的协议,设计重点放在了 IP/TCP/ICMP/UDP/ARP 这些网络层和传输层协议上,保证了代码的通用性和结构的稳定性。正因为存在代码量非常少,占用内存小,硬件处理层、协议栈层和应用层共用一个全局缓存区,不存在数据的拷贝,无需操作系统支持等优点,uIP 协议栈现已被广泛移植到 8/16/32 单片机上使用。

uIP 协议栈相当于一个代码库,通过一系列的函数实现与底层硬件和上层应用程序进行通讯,对于整个应用系统来说它内部的协议组是透明的,从而增加了协议的通用性^[9]。uIP 协议栈与系统底层和上层应用之间的关系图 3 所示。

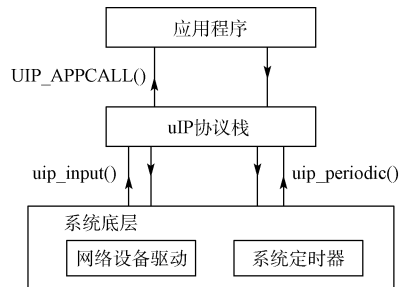


图 3 uIP 协议栈框架

从图 3 可以看出,uIP 协议栈主要提供了 3 个函数 uip_init()、uip_input()和 uip-periodic()。其中与上层应用程序的接口是 UIP_APPCALL()宏。uip_init()是系统初始化时调用的,主要用于初始化协议栈;uip_input()函数则会根据接收到数据包首部的协议进行处理,并在需要时调用应用程序;uip_periodic()函数会在 10 s 内被调用一次,用于驱动所有 uIP 内部需要时钟的事件,当定时到期时,都会进行 ARP 老化处理或者更新所有 TCP 连接状态;UIP_APPCALL()定义为应用层处理函数。

3 HTTP 协议与 JSON 格式

HTTP 协议是一种超文本传输协议,属于应用层协议,主要通过传输层 TCP 协议在服务器与客户端之间实现数据传输。它是一个客户端和服务器端请求与应答的标准,当客户端与服务器建立连接后,发送一个请求给服务器;服务器接到请求后,给予相应的响应信息。

HTTP 协议具有如下特点。

- 1)简单快速:客户向服务器请求服务时,只需传送请求方法和路径。
- 2)灵活:HTTP 允许传输任意类型的数据。仅仅只需设置 Content-Type 标记即可。
- 3)无连接:无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求,并收到客户的应答后,即断开连接。采用这种方式可以节省服务器的资源消耗。

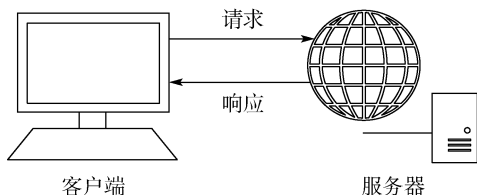


图 4 HTTP 协议请求与响应

4) 无状态: 服务器记录不会对客户端的请求历史进行, 可以节省服务器的响应时间, 支持多并发。

JSON 是一种轻量级的数据交换格式, 数据存在两种结构: 对象、数组。对象在 JSON 中表示为“{ }”括起来的内容, 数据结构为 key-value 的键值对结构, 比如 { "name": "Jim" } 表示某个人姓名为“Jim”, 基本上望文识意, 非常容易理解; 数组则表示为“[]”括起来的内容, 比如 ['Jim', 'Anna'] 代表数组中存在“Jim”与“Anna”两个字符串, 其结构基本和 C 语言中的数组表示一样。经过对象、数组 2 种结构就可以组成复杂的数据结构了。

4 uIP 移植与 HTTP 实现

4.1 ENC28J60 硬件底层驱动

uIP 移植的首要任务是编写网卡驱动程序, 即通过 SPI 接口驱动以太网芯片 ENC28J60^[10-11]。需要设置的部分包括发缓冲区的大小和起始地址、设置以太网过滤器、配置 MAC 层以及物理层等。

4.1.1 ENC28J60 发送数据

ENC28J60 内的 MAC 控制器在发送数据时会自动生成前导符和帧起始界定符, 还可以根据配置生成填充和 CRC 字段。主控制器必须生成其它的帧字段, 包括报头、目的地址、源地址, 并将它们写入缓冲存储器, 以待发送。发送过程如图 5 所示。

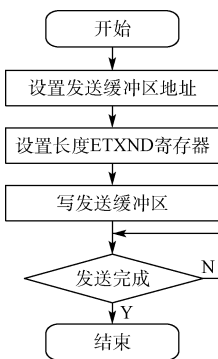


图 5 ENC28J60 发送流程

4.1.2 ENC28J60 接收数据

初始化接收缓冲区完成以后, 必须正确配置接收过滤器接收以太网数据包。开启接收使能, ENC28J60 收到一个完整的以太网数据包以后, 过滤器会自动进行过滤, 未经

过滤的数据包会写入接收缓冲器, 同时向处理器发出通知。接收过程如图 6 所示。

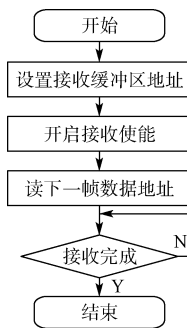


图 6 ENC28J60 接收流程

4.2 uIP 移植

实现了以太网芯片驱动以后, 将驱动程序添加到 Zstack 的 hal 硬件驱动文件夹内。要实现 uIP 协议栈的基本功能, 还需要 periodic 时钟的驱动。在 ZStack 中添加新任务 UipAppTask, 该任务初始化时完成协议栈的基本设置后, 启动一个定时器作为 uip_periodic() 函数的驱动, 每当定时器到期后就完成 APR 老化以及 TCP 连接状态的更新。UipAppTask 初始化流程图如图 7 所示。

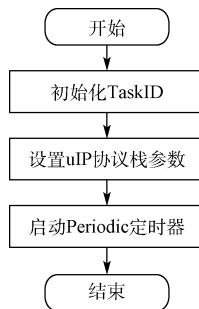


图 7 UipAppTask 初始化

将 uip-1.0 主循环部分代码进行封装, 将其加入到 ZStack 硬件层的 Hal_ProcessPoll 中, 保证 ZStack 可以实现 uip 协议栈数据的处理。

```

void uIP_ZStack_Poll(void)
{
    uip_len = tapdev_read();
    if(uip_len > 0) {
        if(BUF -> type == htons(UIP_ETHTYPE_IP)) {
            uip_arp_ipin();
            uip_input();
            if(uip_len > 0) {
                uip_arp_out();
                tapdev_send();
            }
        }
        else if(BUF -> type == htons(UIP_ETHTYPE_
    
```

```

ARP)) {
    uip_arp_arpin();
    if(uip_len > 0) {
        tapdev_send();
    }
}
}

```

此外,由于CC2530内存限制,为实现ZStack与uIP协议栈的稳定运行,还需对ZStack与uIP协议栈进行裁剪配置。具体宏配置如表1所示。

表1 ZStack与uIP配置

ZStack-2.4.0-1.4.0	uIP-1.0
	MAX_CONNECTIONS = 2
INT_HEAP_LEN = 1536	MAX_LISTENPORTS = 1
	BUFFER_SIZE = 350

4.3 HTTP协议的POST封装

标准HTTP协议提供了GET、POST、HEAD、PUT等方法,文中仅实现了POST请求。图8为HTTP报文的基本格式。

将移植以后的uIP协议栈配置为TCP客户端模式,目标端口设置为80端口,每当需要发送数据时,先将数据按照HTTP协议的POST请求进行封装^[12],同时将POST



图8 HTTP报文格式

参数打包为JSON格式,然后调用uip_send()函数进行发送。主要发送代码如下:

```

//封装POST请求
sprintf(BUF, "POST %s%s HTTP/1.1\r\n\r\n"
Host: %s\r\n\r\n"
User-Agent: ENC28J60\r\n\r\n"
Content-Type: application/x-www-form-urlencoded;
charset=gb2312\r\n\r\n"
Connection: close\r\n\r\n"
ContentLength: %d\r\n\r\n\r\n%s", HTTP_PATH,
PATH_str[path_type], \
HOST_NAME, strlen(gatewaydata), gatewaydata);
uip_send(BUF, strlen(BUF));

```

通过Wireshark软件抓取通信过程,可以看到由ENC28J60发送的POST请求,同时数据格式为JSON格式。结果如图9所示。



图9 获取POST请求数据报文

5 温湿度监测验证

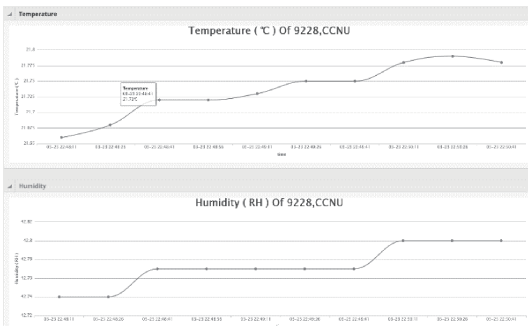
为了验证HTTP协议与JSON格式在ZStack中的正确应用,采用3块CC2530开发板设计了温湿度监测系统。3块开发板的角色分别设定为ZigBee协调器、路由器、终端设备,Zstack网络拓扑为Mesh网络。终端设备通过SHT20温湿度传感器^[13]每15s采集环境温度湿度上传给协调器;路由器和终端设备同时保持1min/次的频率向协调器发送在线心跳包。协调器搭配ENC28J60模块组成ZigBee网关,每当接收到终端设备或者路由器的数据,将数据打包为JSON格式并通过HTTP协议POST请求发

送到PC端PHP后台。

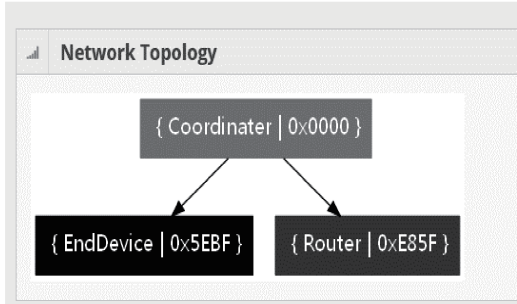
PHP后台会将数据插入到MySQL数据库,同时会在WEB前端实时绘制温湿度曲线以及ZigBee网络的拓扑结构。

6 结论

将uIP协议栈移植到ZStack中,使CC2530直接作为ZigBee以太网网关,而无需借助其它MCU即可无缝接入互联网。创新性的采用HTTP与JSON传输数据,在确保安全可靠的前提下,使数据格式统一,降低了后台解析数据的难度。通过对温湿度监测验证了HTTP协议与



(a)



(b)

图 10 (a)温湿度曲线;(b)ZigBee 网络拓扑

IP stack-interconnecting smart objects with IP-Chapter 13 [C]. Interconnecting Smart Objects with IP, 2010:167-182.

- [6] 李佳, 谢琦, 王庆华. 基于网关的 ZigBee 网络与 Internet 互联框架[J]. 计算机工程与设计, 2012, 33(9):3332-3336.
- [7] 张利, 卞晨, 马学森, 等. 基于小波分析的汽车振动舒适性评价方法[J]. 电子测量与仪器学报, 2014, 28(4):352-359.
- [8] 张大伟, 陈佳品, 冯洁, 等. 面向准危重病病人的区域化无线监护系统研制[J]. 仪器仪表学报, 2014, 35(1):74-81.
- [9] 楼亮亮, 金彦亮, 周苗. TCP/IP 协议在 ZStack 中的移植与应用[J]. 烟台大学学报: 自然科学与工程版, 2016(1).
- [10] 张庆辉, 马延立. STM32F103VET6 和 ENC28J60 的嵌入式以太网接口设计[J]. 单片机与嵌入式系统应用, 2012, 12(9):23-25,32.
- [11] 刘宗铭, 卓振泰, 何明华. 基于 ENC28J60 的以太网接口的设计与实现[J]. 电子器件, 2013, 36(6):919-923.
- [12] 刘龙启, 李银. 基于 LabVIEW 的以太网数据监听与通信[J]. 国外电子测量技术, 2012, 31(7):62-65.
- [13] 周海鸿, 周嘉奉. 基于 ZigBee 技术的温湿度监测系统[J]. 国外电子测量技术, 2015, 34(7):75-79.

JSON 格式在 ZStack 中的应用。

参考文献

- [1] 黄真. 基于 Z-Stack 的 WSN 节点中断程序设计[J]. 单片机与嵌入式系统应用, 2011, 11(7):79-81.
- [2] 王涛. HTTP 协议技术浅析[J]. 中国新技术新产品, 2013(22):14-14.
- [3] 高静, 段会川. JSON 数据传输效率研究[J]. 计算机工程与设计, 2011, 32(7):2267-2270.
- [4] 王魁生, 王晓波. 利用 JSON 进行网站客户端与服务端数据交互[J]. 软件导刊, 2010, 9(3):147-149.
- [5] VASSEUR J P, DUNKELS A. uIP — A lightweight

作者简介

卢少军, 1990 年出生, 华中师范大学物理科学与技术学院硕士研究生, 主要研究方向为嵌入式系统及应用软件开发。
E-mail: 450276860@qq.com

刘守印, 1964 年出生, 华中师范大学物理科学与技术学院教授, 主要研究领域为 B3G、4G 相关理论研究、嵌入式与物联网。

E-mail: syliu@phy.ccnu.edu.cn