

用 STM32 和 ESP8266 实现的可扩展物联网系统*

常欣, 王琦

(武汉科技大学 信息科学与工程学院, 武汉 430081)

摘要: 在以 STM32 为主控芯片的基础上, 结合 ESP8266 网络芯片, 构建出一个可扩展的物联网设备系统。本系统可灵活接入多种外部设备, 实现了将硬件产生的数据通过软件统一处理为标准网络数据, 使得系统具备硬件兼容能力。本系统较好地解决了简易嵌入式系统在设计和使用时实时性差、兼容困难的问题。

关键词: 物联网; STM32; ESP8266; EDP 协议; OneNET

中图分类号: TP368

文献标识码: A

Scalable IoT System Based on STM32 and ESP8266

Chang Xin, Wang Qi

(College of Information Science and Engineer, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: In the paper, based on STM32 as the main control chip and the ESP8266 network chip, a scalable IoT device system is constructed. The system not only has flexible access to a variety of external devices, but also has certain hardware compatibility capabilities, that is, the data generated by hardware is uniformly processed into standard network data through the software. The system better solves the problem that the simple embedded system has poor real-time performance and compatibility in design and use.

Key words: Internet of Things; STM32; ESP8266; EDP protocol; OneNET

引言

物联网是新一代信息技术的重要组成部分, 通过智能组网、智能传输技术将传统网络的覆盖范围扩展到物与物的级别。当前物联网已发展成为对物体具有全面感知能力, 对信息具有可靠传递和智能处理能力的连接物与物之间的信息网络^[1]。STM32 在原有的工业市场已经具有很好的基础, 在其上发展加入更多的联接与云端的服务具有很大的市场成长空间^[2]。本文在嵌入式应用最为典型 STM32 系列芯片的基础上, 结合使用 ESP8266 网络通信芯片, 通过软硬件设计, 解决了嵌入式系统常见的运行时状态独立、数据传输缺乏实时性的问题, 有效拓展了其联网应用能力。

1 系统框架组成

本系统主要分为三大模块: 数据采集端、控制端、网络端。数据采集端是整个系统的数据入口, 负责搜集传感器数据。网络端的主要功能是在网络上对数据进行展示, 本文使用 OneNET 作为数据接入和管理平台, 该平台同时提供了基于数据认证和保护机制的信息可视化服务, 允许

用户查看系统详细数据, 并在任何时间接收通知^[3]。控制端是本系统的核心, 承担着数据接收、缓存、处理以及发送任务。系统架构如图 1 所示。

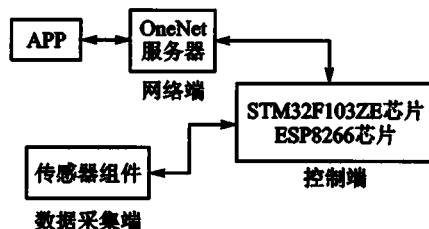


图 1 系统框架图

2 主设备硬件设计

本系统的主要硬件组成如图 2 所示。其中, STM32F103ZE 负责数据处理, 由于其拥有 112 个可用的 I/O 端口, 以及 2 个 I²C 接口和 SPI 接口、3 个 USART 接口、一个 USB 接口和一个 CAN 接口, 每个 I/O 均可接入相应外设^[4]。所以根据需求可快速灵活地接入各种不同的外部设备, 系统可实现的具体功能由所接入传感器的类型决定。ESP8266 负责网络通信, 具体采用了 ESP8266 01S 无线模块, 该模块支持标准的 IEEE802.11 b/g/n 协

* 基金项目: 武汉科技大学研究生教改项目(项目编号: 2016-23)。

议,此外还集成了包括天线开关、电源管理等硬件单元。不仅能解决单个系统中的网络连接问题,也可以避免节点到服务器通信中任何类型的故障^[5]。ESP8266 的传感器及其连接执行器通过网络进行监控和控制,现有设备可以用来加强对应功能并令其更加智能^[6]。集成调试单元主要用于编码阶段的代码调试。状态指示单元用于工作状态监测。

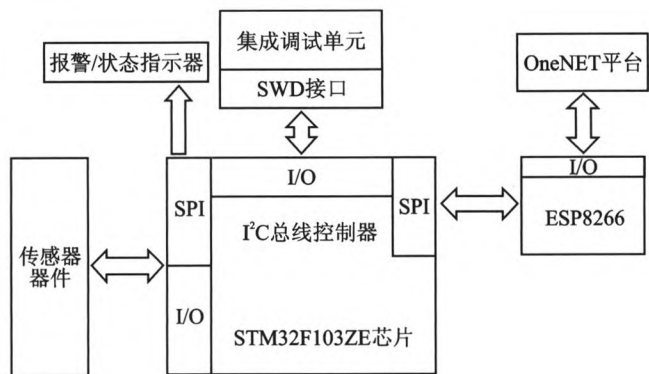


图 2 系统硬件组成图

ESP8266 01S 无线模块与 STM32F103ZE 连接需要 4 个接口,具体设计如图 3 所示。

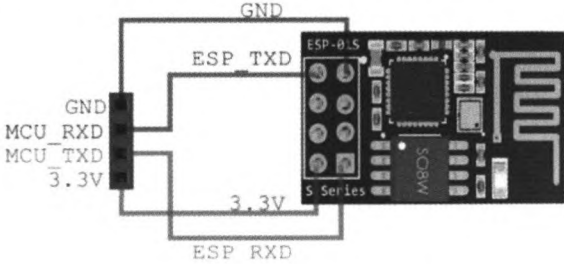


图 3 ESP8266 接线以及实物图

板载电源提供 3.3 V 和 GND 接口。MCU_RXD 接口负责数据上传,MCU_TXD 接口负责数据下载。ESP_TXD 接主控制器芯片的 PA2 口,ESP_RXD 接 PA3 口,二者均被配置为串口 (USART2)。

3 系统软件设计

本系统的硬件程序采用 C 语言编写,软件部分包括硬件驱动、硬件间协调、数据处理以及格式化、数据转发等功能。此外,本系统中用到的 EDP 协议包由已在 GitHub 上开源 OneNET 云平台代码经过适当改进而来,数据传输基于该协议。

3.1 主函数程序设计

本系统的工作过程大致可分为硬件初

始化、建立网络接、采集数据处理并转发三大过程,流程图如图 4 所示。

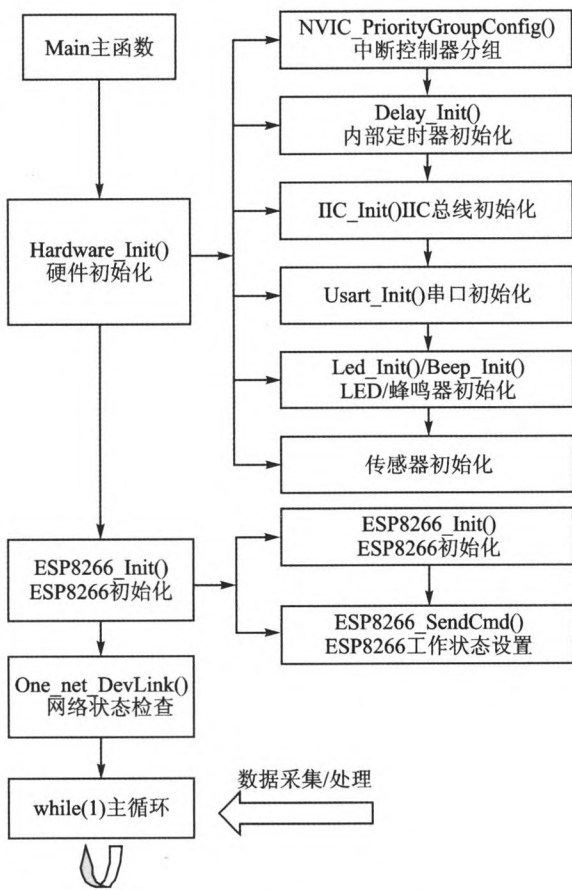


图 4 Main 函数流程图

3.2 ESP8266 程序设计

ESP8266 程序主要完成硬件层面的初始化以及软件层面的网络连接两大功能,其流程图如图 5 所示。

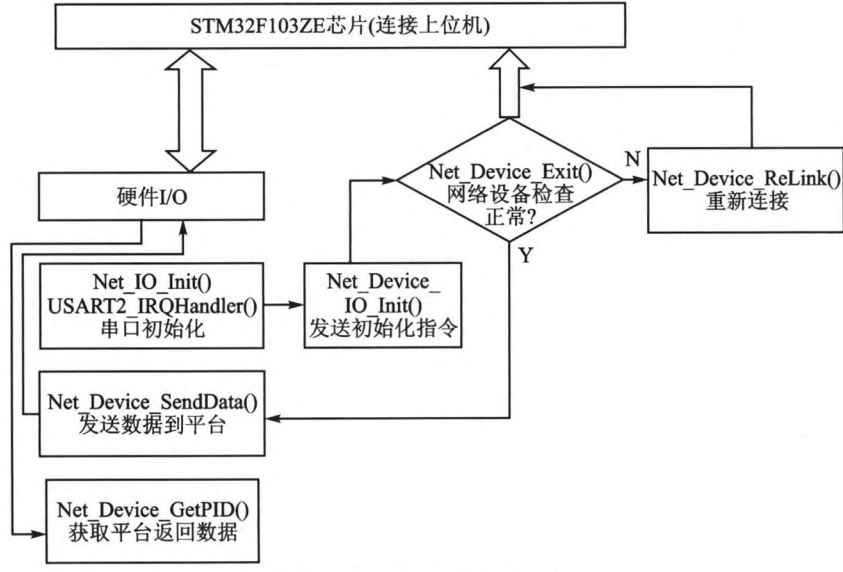


图 5 ESP8266 程序流程图

本文使用的 ESP8266 01S 与 STM32F103ZE 之间的通信采用基于 AT 指令的串口通信方式。串口是计算机上一种非常通用的通信协议,大多数计算机包含两个基于 RS232 的串口,串口同时也是仪器仪表设备通用的通信协议^[7]。其初始化代码如下:

```
void NET_DEVICE_IO_Init(void){
    GPIO_InitTypeDef gpioInitStruct;
    gpioInitStruct.GPIO_Mode=GPIO_Mode_Out_PP;
    gpioInitStruct.GPIO_Pin=GPIO_Pin_0;
    //GPIOA0 被定义为复位引脚
    gpioInitStruct.GPIO_Speed=GPIO_Speed_50MHz;
    GPIO_Init(GPIOA,&gpioInitStruct);
    NET_IO_Init(); //串口初始化
    netDeviceInfo.reboot=0;
}
```

在 I/O 初始化完成后,需要进一步设置相应的 AT 指令,其中最为关键的两条指令如下:

```
#define ESP8266_WIFI_INFO "AT+CWJAP=\"PDCN\","
"\,\"wx960105\"\\r\\n"
#define ESP8266_ONENET_INFO "AT+CIPSTART=\"TCP\","
"\,\"183.230.40.39\",876\\r\\n"
```

AT+CWJAP 指令的功能是连接无线 WiFi 指令,它有两个参数,分别是 SSID 以及对应的密码。只有正常接入网络才可以向网络服务器发送数据。AT+CIPSTART 指令的功能是访问指定网络位置。它有三个参数,分别是通信协议类型、主机地址、端口号。ESP8266 通过上述两条宏指令实现与远程服务器的连接。

3.3 数据处理程序设计

本系统产生的各种数据通过 EDP 协议进行传输。EDP (Enhanced Device Protocol) 即增强设备协议,是 OneNET 平台根据物联网特点专门定制的完全公开的基于 TCP 的协议,被广泛应用到家居、物流、能源以及其他行业应用中^[8]。

数据传送前需要对数据进行格式化处理。数据格式化的主要工作为封装信息属性和文件拆分。封装信息属性环节包括封装协议头和数据头。协议头指派采用对应协议格式传输,数据头向服务器传递诸如数据类型大小等信息,以便服务器正确处理数据。文件拆分传输包含以下几个环节:提取信息-拆分-传输-接收-组装-检查。接收组装和检查的过程由服务器端自动完成。综上所述,数据格式化的流程图如图 6 所示。

数据处理程序不仅完成了协议要求的数据传输前的各种准备工作,并且实现了硬件和数据间的隔离,即任何由硬件产生的数据经过该程序均可被换为服务器可识别的并且可进行网络传输的数据,极大提高了本系统在数据

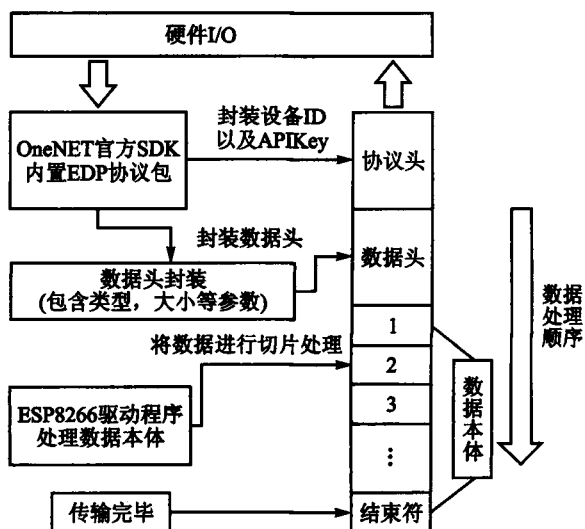


图6 数据处理程序流程图

层面的兼容性。

4 应用示例

4.1 图片传输示例

本系统采用 Flash 芯片 W25Q16,用于储存程序本身和程序运行时所需的数据。在此基础上,构建一个小型化的文件传输服务器,传输原理如图 7 所示。

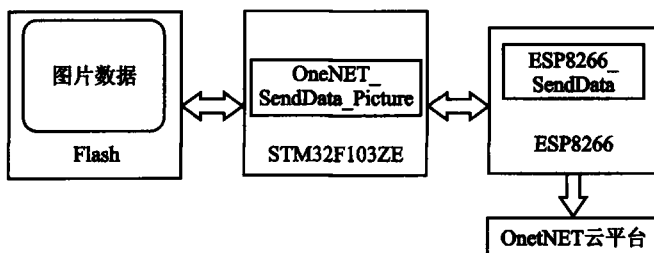


图7 图片上传原理图

准备一张待传输图片,由于 Flash 储存容量的限制,需要对图片进行压缩处理。最后将该图片转存为芯片可以识别的十六进制数组。

```
unsigned char Array[]={
    .....
}
```

转换完成后,将程序烧入 Flash 中并启动系统,同时打开 OneNET 手机 APP,即可看到已上传的图片,如图 8 所示。

4.2 温湿度监测示例

本文演示的第二个功能是基于 DHT22 温湿度传感器的环境温度监测系统。温湿度采集传感器采用常见的 DHT22 芯片,其是一种电容式可输出数字信号的温湿度传感器,它



图8 图片上传成功界面

包括湿度感知单元、NTC 温度感知单元,以及一个 8 位的微型处理器^[9]。

在完成硬件设计与搭建、程序设计以及平台设置之后,本系统即可实现实时监测室内温度湿度并上传至网络的功能。如图 9 所示,打开 OneNET 旗下手机客户端设备云即可实时查看对应设备下对应数据点的实时数据。打开对应网页即可看到一段时间内的温湿度变化趋势以及当前温湿度值,对应界面如图 10 所示。



图9 手机端实时监测界面

结 语

本文使用了 ESP8266 芯片,使原有的 STM32 系统具备了一定的物联网能力。在此基础之上,通过一定的

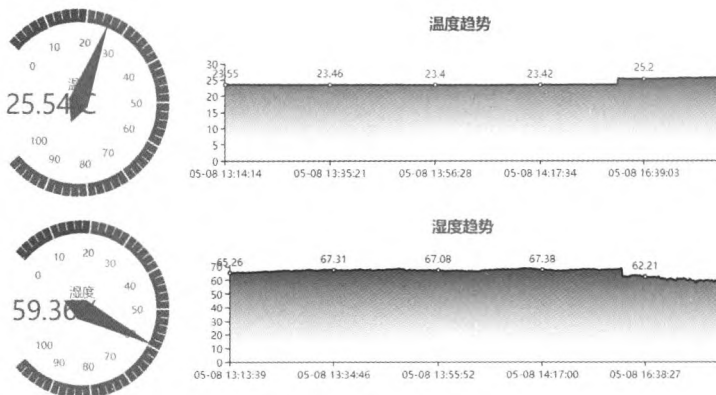


图10 网页端数据流展示界面

编程设计,使得本系统具备良好的硬件兼容性,从而有效解决了简易嵌入式系统在设计和使用中实时性差、兼容困难的问题。

参考文献

- [1] 姜艳. 物联网的发展趋势及应用前景[J]. 企业改革与管理, 2017(22):48.
- [2] 王莹,王金旺. 工业物联网浪潮下的嵌入式系统[J]. 电子产品世界, 2017(11):20-22,25.
- [3] Marques G, Pitarm R. An Indoor Monitoring System for Ambient Assisted Living Based on Internet of Things Architecture [J]. Int J Environ Res Public Health, 2016, 13 (11):1152.
- [4] 崔琳,朱磊,刘小龙,等. 基于 STM32F407 的以太网通信模块设计[J]. 计算机测量与控制, 2018(1):260-263.
- [5] Thaker T. ESP8266 based implementation of wireless sensor network with Linux based web - server[C]//Colossal Data Analysis and NETWORKING. IEEE, 2016:1-5.
- [6] Kodali R K, Soratkal S R. MQTT based home automation system using ESP8266[C]//Humanitarian Technology Conference. IEEE, 2017:1-5.
- [7] 张庆法. 基于 IrDA 的串口通信设计[J]. 微计算机信息, 2006, 22(33):169-171.
- [8] 尤琦涵,陈兆仕,张沁. OneNET 云平台 WiFi 远程控制的智能教室系统[J]. 单片机与嵌入式系统应用, 2017(10):69-73.
- [9] Hernándezrivera D, Rodríguezroldán G, Moramartínez R, et al. A Capacitive Humidity Sensor Based on an Electrospun PVDF/Graphene Membrane [J]. Sensors, 2017, 17 (5): 2-11.

常欣(本科生),电气工程及其自动化专业;王琦(讲师),主要研究方向为储能系统暂态特性。

(责任编辑:薛士然 收稿日期:2018-09-03)