ESP8266 在智能家居监控系统中的应用*

范兴隆

（山东华宇工学院，德州 253034）

摘要：ESP8266 是乐鑫公司生产的低功耗 WiFi 芯片，内置 32 位 CPU，能够独立运行，也可以作为从机搭载于其他主机 MCU 运行，可以广泛应用于智能家居、工业无线控制、无线传感器等领域。本文以一套基于 WiFi 组网的智能家居监控系统设计为例，介绍 ESP8266 的 SDK 应用开发。

关键词：WiFi；ESP8266；DHT11；E4A

中图分类号：TN92　　**文献标识码**：A

Application of ESP8266 in Intelligent Home Monitoring System

Fan Xinglong

（Shandong Huayu University of Technology，Dezhou 253034，China）

Abstract：The ESP8266 is produced by ESPRESSIF System，which is low-power consumption Wi-Fi chip，and integrates 32-bit MCU. It can run independently，and also can be used as a slave of the other host MCU. It can be widely used in smart home，industrial wireless control，wireless sensors and other fields. In this paper，taking the smart home monitoring system based on WiFi network as an example，the ESP8266 software development is introduced.

Key words：WiFi；ESP8266；DHT11；E4A

引　言

随着 ZigBee 技术、蓝牙技术及 WiFi 技术的不断成熟与普及，基于各种组网方式的智能家居监控系统越来越多，给人们的生活带来了极大的方便。其中，WiFi 组网以其方便与有线以太网络整合、组网的成本低等优势，逐渐受到人们的推崇。目前国内外 WiFi 芯片生产厂商越来越多，芯片性能越来越好，其中 ESP8266 就是一款性价比较高的低功耗 WiFi 芯片。本文介绍了一种基于 WiFi 组网的家居监控系统，该系统利用功能各异的多个 ESP8266 模块组建无线局域网，使用手机端 APP 对网络中各个模块的工作进行监控，不仅可以实现灯具、窗帘等的无线开关，还可以对室内环境温度、湿度及空气成分等指标进行检测。通过对该系统的介绍，重点讲述 ESP8266 的 SDK 开发过程。

1　总体设计

WiFi 组网框图如图 1 所示，本系统采用 BSS（Basic Service Set，基本服务集）模式进行 WiFi 组网，建立 1 个 softAP（无线接入点，是一个无线网络的中心节点）和多个与其关联的 station（无线终端是一个无线网络的终端），手机可以以 station 身份连接 softAP。系统中 softAP 模块、station 模块均基于 ESP8266 进行搭建，各 station 模块实现不同的功能，如温湿度检测、灯具开关等，softAP 模块收集各 station 模块的数据，手机端利用 APP 对 softAP 模块进行访问，查询相关状态数据并进行控制指令下达。

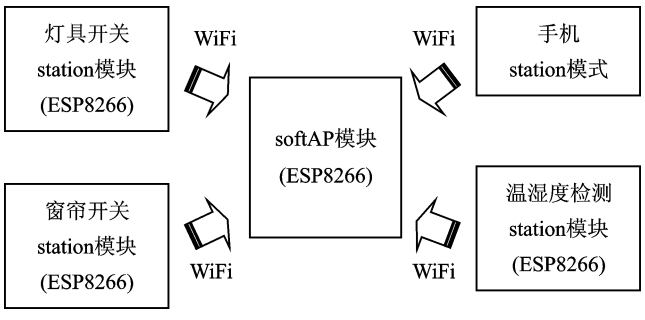

图 1　WiFi 组网框图

2　硬件设计

2.1　ESP8266 简介

ESP8266 是一个完整且自成体系的 WiFi 网络解决

* 本文得到山东省高等学校青年骨干教师国内访问学者项目经费资助。

方案，能够独立运行，也可以作为从机搭载于其他主机MCU运行。ESP8266在搭载应用并作为设备中唯一的应用处理器时，能够直接从外接闪存中启动。内置的高速缓冲存储器有利于提高系统性能，并减少内存需求。另外一种情况是，ESP8266负责无线上网接入承担WiFi适配器的任务，此时可以将其添加到任何基于微控制器的设计中。ESP8266高度片内集成，包括天线开关balun、电源管理转换器，因此仅需极少的外部电路，且包括前端模组在内的整个解决方案在设计时将所占PCB空间降到最小。

2.2 ESP8266模块电路设计

ESP8266高度集成的特点使得其外围设计非常简单容易。乐鑫官方提供的基本模组电路如图2所示。模组除主芯片外只需要1个无源晶振、1个SPI Flash及若干电阻、电容、电感。射频部分实现全内部集成，并且内部带有自动校准功能。本设计选用了深圳安信可公司设计的ESP-12模块，该模块采纳了图2所示的电路结构，采用PCB天线，经过匹配设计，空旷环境下传输距离可达到400 m左右，所有I/O口引出，带金属屏蔽壳，通过FCC&CE认证。

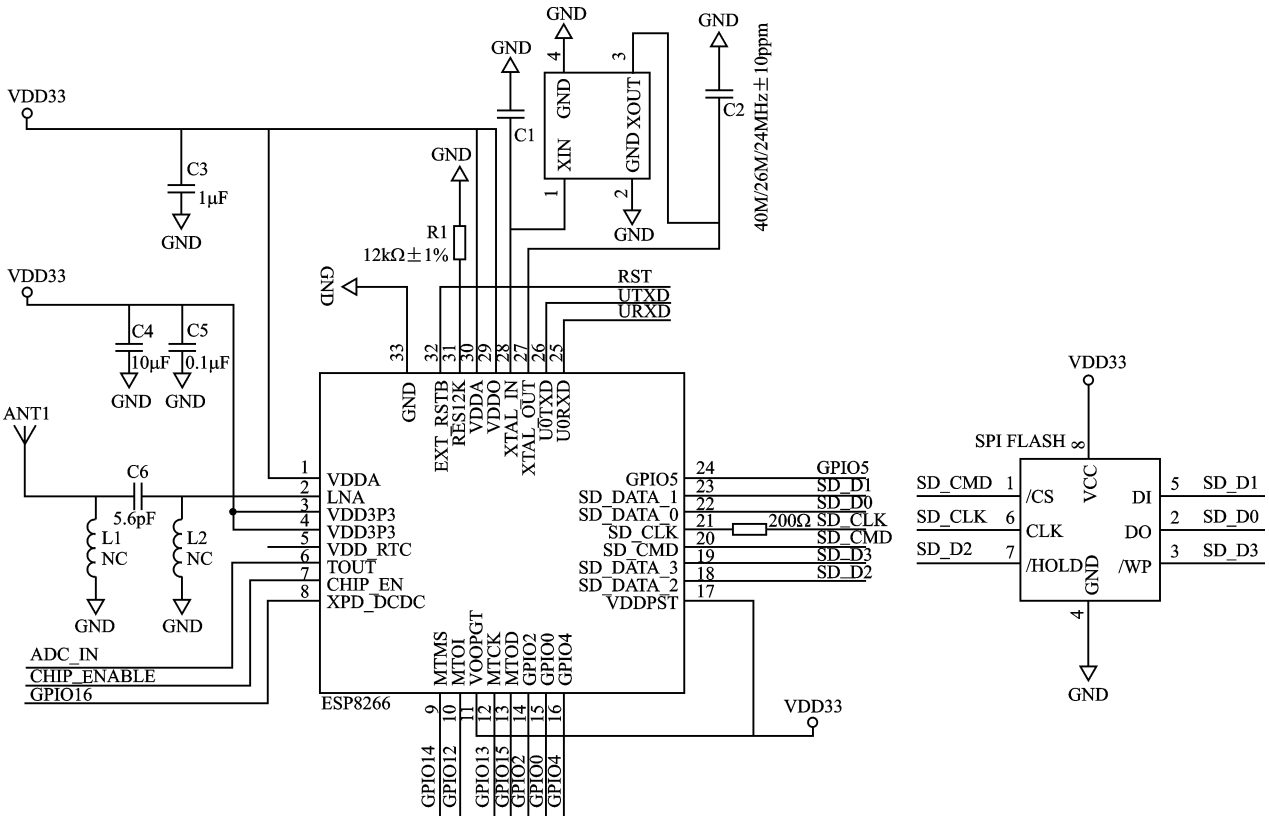

图2 ESP8266模组电路图

本设计中softAP模块无其他外扩电路，灯具开关station模块、窗帘开关station模块需外接继电器实现被控对象控制，温湿度检测station模块需外接DHT11温湿度传感器。图3为DHT11的接线图，DATA为数据总结引脚，采用单总线数据格式。

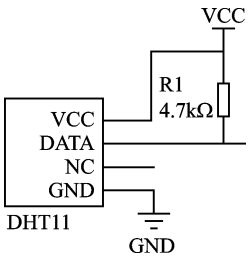

图3 DHT11接线图

DHT11数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器，它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性与卓越的长期稳定性。传感器包括一个电阻式感湿元件和一个NTC测温元件，并与一个高性能8位单片机相连接。

3 软件设计

3.1 ESP8266SDK软件包简介

基于ESP8266物联网平台的IoT SDK为用户提供了一个简单、快速、高效开发物联网产品的软件平台。SDK为用户提供了一套数据接收、发送的函数接口，用户不必关心底层网络，如WiFi、TCP/IP等的具体实现，只需要专注于物联网上层应用的开发，利用相应接口完成网络数据的收发即可。

ESP8266物联网平台的所有网络功能均在库中实现，对用户不透明。用户应用的初始化功能可以在user_main.c中实现。void user_init(void)是上层程序的入口函数，给用户提供一个初始化接口，用户可在该函数内增加

硬件初始化、网络参数设置、定时器初始化等功能。为方便二次开发，SDK 提供了较为丰富的 API 接口，接口具体信息可参考 ESP8266__SDK__Programming Guide。

3.2 ESP8266 IDE 简介

ESP8266 IDE 为安信可公司开发的 ESP8266 模块编译平台，具有免安装、纯绿色、无需虚拟机、Windows 系统直接运行、IDE 界面、编辑和编译一体化、Eclipse 编译后直接生成固件功能。

3.3 softAP 模块程序设计

该系统中存在多个 station 模块，在系统正常工作时，需 softAP 模块同时与多个 station 模块进行通信，为保证信息的可靠传输，网络通信采取 TCP/IP 协议。softAP 模块作为系统的中心节点，设置为 TCP 服务器，采用 DHCP 方式，为各 station 模块动态分配 IP 地址，各模块以客户端的身份与其建立连接。在连接建立完成后，softAP 模块负责监听各模块的工作状态，同时收集相关模块发送来的数据信息，并进行分类存储。

softAP 的程序入口函数 user_init( ) 工作流程图如图 4 所示。系统初始化主要完成 μs 级定时器重新初始化、串口波特率设置、GPIO 引脚功能选择；μs 级定时器主要为实现循环工作任务而设置；WiFi 工作模式设置为 soft-AP；soft-AP 接口配置主要包括 ssid 设置、password 设置；在 WiFi event 处理函数中进行 WiFi 工作状态查询与串口输出。

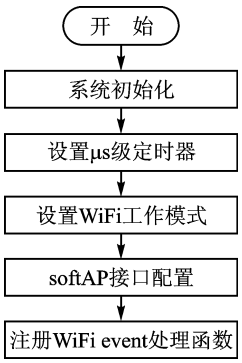

图 4 softAP 程序入口函数工作流程图

下为 user_init( ) 主要程序指令及注释：

```
system_timer_reinit();
//重新初始化定时器，当需要使用 μs 级定时器时调用
uart_init(115200,115200);
//双 UART 模式，两个 UART 波特率初始化
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO0_U,FUNC_GPIO0);//引脚功能选择
os_timer_setfn(&connect_timer1,timer1,NULL);
//设置 μs 定时器回调函数
os_timer_arm_us(&connect_timer1,1000,1);
//使能 μs 级定时器
wifi_set_opmode(0x02);//设置 WiFi 工作模式 softAP
wifi_softap_get_config(&config);
os_memcpy(config.ssid," ESP8226",strlen(" ESP8226"));
                                        //设置 ssid
os_memcpy(config.password," 12345678",strlen(" 12345678"));
                                        //设置 password
config.ssid_len=strlen(" ESP8226");
wifi_softap_set_config(&config);
                    //设置 WiFi softAP 接口配置，并保存到 Flash
wifi_set_event_handler_cb(wifi_handle_event_cb);
                                        //注册 WiFi event 处理回调
```

WiFi event 处理函数主要完成 station 连接、DHCP 配置、TCP 连接及无线数据收发等任务，其工作流程如图 5 所示。在 WiFi event 处理过程中，首先确定是否有 station 连接，若连接成功进而判断 DHCP 配置是否完成，在前述两个条件均满足的情况下，进行 TCP 网络连接参数 espconn 设置，并注册 TCP 网络连接回调函数及建立 TCP 侦听。在接收到建立连接的 TCP 客户端数据后，在连接回调函数中首先进行数据类型判别，然后根据不同类型数据进行相应处理，主要实现信息、状态类数据的存储和控制指令的发送。

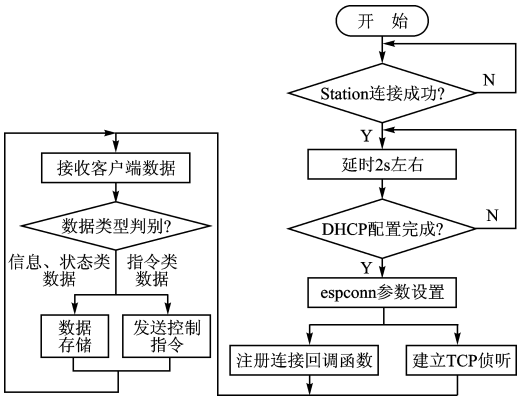

图 5 WiFi event 处理函数工作流程图

以下为建立 TCP 连接的程序指令及注释：

```
user_tcp_conn.type=ESPCONN_TCP;  //选择 TCP 连接方式
user_tcp_conn.state=ESPCONN_NONE;  //状态
user_tcp_conn.proto.tcp=(esp_tcp *)os_zalloc(sizeof(esp_tcp));
                                        //分配内存空间
os_memcpy(user_tcp_conn.proto.tcp->local_ip,local_ip,4);
                                        //存放本地 IP
os_memcpy(user_tcp_conn.proto.tcp->remote_ip,remote_ip,4);
                                        //存放远端 IP
user_tcp_conn.proto.tcp->local_port=8080;
                                        //本地端口设置
user_tcp_conn.proto.tcp->remote_port=remote_port;
                                        //远端端口设置
espconn_regist_connectcb(&user_tcp_conn,user_tcp_connect_cb);
//注册连接成功的回调函数
espconn_regist_reconcb(&user_tcp_conn,user_tcp_recon_cb);
//注册连接失败的回调函数
espconn_accept(&user_tcp_conn);//创建 TCP server，建立侦听
espconn_regist_time(&user_tcp_conn,60,0);
```

//设置 TCP server 连接超时时间

3.4 station 模块程序设计

station 模块作为系统的终端节点，承担环境信息监测、负载控制及与 softAP 模块进行通信等任务，在网络通信中设置为 TCP 客户端。

各 station 模块的 user_init( )函数程序工作流程如图 6 所示。系统初始化、μs 级定时器设置与 softAP 模块的设置相同，WiFi 工作模式设置为 station，下面的工作应为扫描获取可连接的 AP 信息，但该功能的接口函数需在系统初始化完成回调函数中调用。因此，user_init( )函数的最后一步为注册系统初始化完成的回调函数。

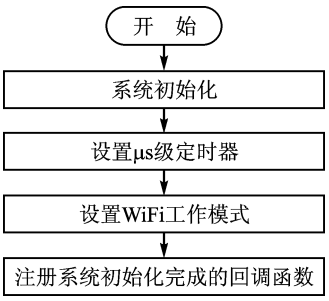

图 6 station 程序入口函数工作流程图

```
wifi_set_opmode(0x01);//设置 WiFi 工作模式 station
system_init_done_cb(to_scan);
//在 user_init 中调用，注册系统初始化完成的回调函数
void to_scan(void) {wifi_station_scan(NULL,scan_done); }
//扫描获取所有可用的 AP 信息，并接入指定 softAP
```

scan_done()为 wifi_station_scan 的回调函数，在该函数中需列出扫描到的 AP 信息，然后根据 softAP 模块的 ssid、password 设置 station 接口配置参数，并接入指定 AP。以下为 scan_done()函数中的主要程序指令：

```
os_memcpy(&stationConf.ssid, SSID, 32);
os_memcpy(&stationConf.password, PASS, 64);
wifi_station_set_config_current(&stationConf);
//设置 station 接口的配置参数
wifi_station_connect();//station 接口连接 AP
```

在函数 wifi_station_connect( )执行之后，需延时 2 s 左右查询 station 接口连接 AP 的状态，只有 station 成功连接到指定 AP，同时获取了 AP 分配的 IP 地址，方可进行 TCP 网络配置及连接。

```
status=wifi_station_get_connect_status();
                                    //查询 station 接口连接 AP 的状态
if(status==STATION_GOT_IP){
    struct ip_info info;//查询 IP 地址的结构体
    wifi_get_ip_info(STATION_IF,&info);
    station_init((struct ip_addr *)remote_ip,&info.ip,8080);
    //获取 IP、端口号，调用网络连接初始化函数
    return;
}
```

station_init( )为网络连接初始化函数，主要进行网络连接参数 espconn 设置，注册 TCP 网络连接回调函数及完成与 TCP 服务器的连接。TCP 网络连接成功后，接收的服务器数据在网络连接回调函数中进行处理，实现 station 模块的相应功能。

3.5 DHT11 数据采集

DHT11 模块的 DATA 端口用于与 ESP8266 模块之间的通信和同步，采用单总线数据格式，一次通信时间为 4 ms 左右，数据分小数部分和整数部分共 40 位。ESP8266 模块发送一次开始信号后，DHT11 从低功耗模式转换到高速模式，等待主机开始信号结束后，DHT11 发送响应信号，送出 40 位的数据，并触发一次信号采集。ESP8266 模块可选择读取部分数据，系统设计 100 ms 读取一次 DHT11 数据。图 7 为 DHT11 的通信时序。

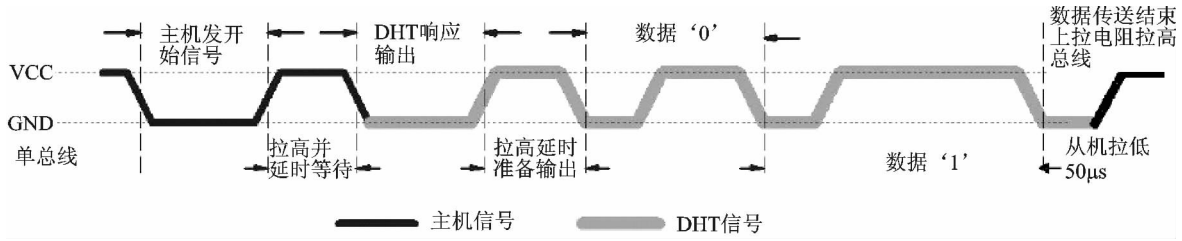

图 7 DHT11 的通信时序

需要注意的是，ESP8266 不带操作系统，由于是单线程，任何事件都不能长期占用 CPU(一般不得超过 500 ms)，如果一个事件占用 CPU 不退出，将导致看门狗的喂狗函数无法执行，系统重启。因此，在 ESP8266 周期性的查询功能最好使用定时器，如需在定时器的执行函数中调用 while、for 等函数进行延时或循环操作，占用时间请勿超过 10 ms，这也是在 softAP 模块中采用 μs 定时器实现循环工作任务的原因。

3.6 无线通信协议制定

本系统 station 模块包含传感器模块、执行模块及手机终端设备等多种类型，既要实现传感器数据的获取，也要实现控制执行器，同时还要与手机终端进行数据交换。由于 station 模块及设备种类较多且为了利于扩展同种类的设备，需要相应的通信协议进行规范。协议格式如表 1 所列，模块类型及数据格式对应关系如表 2 所列。

表 1　协议格式

模块编号	模块数据	扩展数据	在线状态
1 * sizeof (unsigned char)	1 * sizeof (unsigned long)	1 * sizeof (unsigned char)	1 * sizeof (unsigned char)

表 2　模块类型及数据格式对应关系

模块名称	模块编号	模块数据
手机终端	0x01	0x01－温度值读取；0x02－湿度值读取 0x03－灯具打开；0x04－灯具关闭 0x05－窗帘打开；0x06－窗帘关闭
温湿度检测模块	0x02	HH HL TH TL；0xFF－命令失败
灯具开关模块	0x03	0x01－开；0x02－关；0xFF－命令失败
窗帘开关模块	0x04	0x01－开；0x02－关；0xFF－命令失败

4　手机端软件设计

本监控系统中手机是非常重要的 station 设备，要实现与 softAP 模块的连接，并查询 softAP 模块中存储的数据，还要将数据显示在手机上，以便用户了解系统工作情况，同时根据用户操作发送控制指令。为实现手机端的功能，采用易安卓软件设计了一个简单的手机 APP，可以运行在安卓系统的移动终端上。图 8 为 APP 用户界面，使用时首先让手机 WiFi 连接 softAP 模块，然后启动 APP，点击连接按键，当如图 8 所示显示“连接成功！”时，手机即可实现系统运行的监控。

图 8　APP 用户界面

结　语

通过以上系统设计介绍，可以简单了解 ESP8266 SDK 开发的流程，不难看出 ESP8266 除了作为 WiFi 透传模块使用外，独立运行也具有较强的功能，可以实现系统的低功耗控制，并且通过连接无线路由器，可以实现物联网系统的云端访问与控制。因此，该芯片将在移动设备开发、可穿戴电子产品设计及物联网应用中发挥越来越重要的作用。

参考文献

[1] Espressif Systems. ESP8266 SDK 编程手册 V1. 3. 0，2015.
[2] Espressif Systems. ESP8266 SDK User Manual V1. 3. 0，2015.
[3] 易安卓开发公司. 易安卓使用指南，2015.
[4] 雁凌电子. DHT11 温湿度模块使用说明[EB/OL]. [2016-04-22]. http://ylelectronic.taobao.com.
[5] 李菲. 智能家居技术浅谈[J]. 科技致富向导，2015(5)：180.
[6] 梁永恩，万世明. 基于 S3C6410 的智能家居控制系统的设计[J]. 计算机与数字工程，2014(6)：1104-1107.

范兴隆（讲师），研究方向为自动检测与控制。

（责任编辑：杨迪娜　收稿日期：2016-04-22）

51 且软件设计时可以淡化底层硬件，便于扩展维护。采用无线串口通信可以满足如车辆定位、液位测量、工业控制多种测距场合。本系统硬件成本低、人机界面直观、测量精度高，具有较高的推广使用价值。

参考文献

[1] 兰羽. 具有温度补偿功能的超声波测距系统设计[J]. 电子测量技术. 2013，36(2)：85-87.
[2] 赵海鸣，卜英勇，王纪婵，等. 一种高精度超声波测距系统的研制[J]. 矿业研究与开发，2006，26(3)：62-65.
[3] 张攀峰，王玉萍，张健，等. 带有温度补偿的超声波测距仪的设计[J]. 计算机测量与控制，2012，20(6)：1717-1732.
[4] 何凡，沈谅平，王浩. 基于温度补偿的超声测距系统设计[J]. 物联网技术，2016(2)：11-16.
[5] 胡延苏，高昂. 超声波测距误差分析及校正研究[J]. 计算机测量与控制，2015，23(8)：2820-2823.
[6] 李光明，孙英爽，党晓娟. 基于 LabVIEW 和 Arduino 的远程监控系统设计与实现[J]. 计算机测量与控制，2015，23(10)：3522-3528.
[7] 刘卫国，王红彬. 基于 nRF24L01＋与 Arduino 的超声波测距系统设计[J]. 电子设计工程，2015，23(22)：150-152.
[8] 沈金鑫. Arduino 与 LabVIEW 开发实战[M]. 北京：机械工业出版社，2014：123-127.
[9] 邹杨，石红瑞. 基于 LabVIEW 的 Tripod 机器人视觉处理和定位研究[J]. 机电工程，2016，33(4)：448-452.
[10] 陶明超，何璐璐，侯佩臣，等. 基于 LabVIEW 的显微镜自动控制设计[J]. 计算机测量与控制，2016，24(1)：102-104.

朱志强（硕士），研究方向为电子通信及嵌入式系统。

（责任编辑：杨迪娜　收稿日期：2016-05-03）