

SWIFI 模块使用教程

Hackathon OMNI-Lab

一、CDUINO 扩展板

扩展板接口如下图：

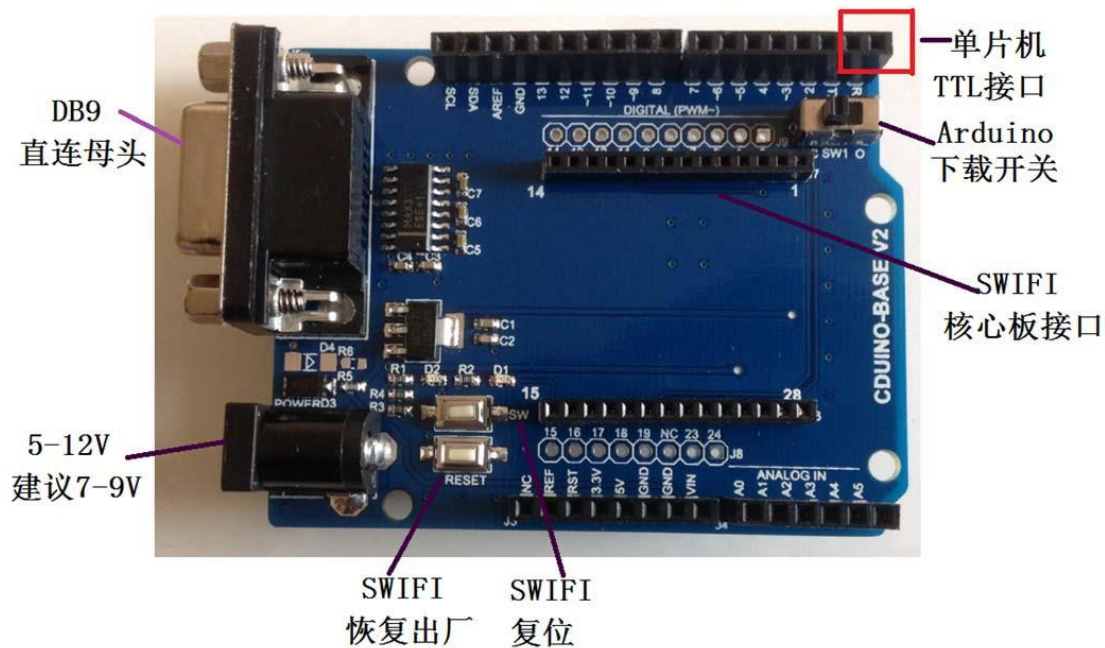


图 1.1 扩展板接口

其中：

电源部分：DC5.5*2.1 接口，可以兼容 5-12V1A，建议 7-9V1A 适配器。SWIFI 核心板功耗极限功耗为 3.3V*180mA，请根据这个标准选择您的电源功率。

DB9 直连母头：连接RS232 公口设备，可以直接和SWIFI 通信。

单片机TTL 接口：包括RXD TXD 两个管脚，分别连接单片机的RXD TXD 管脚，切莫交叉。管脚电平为3.3V TTL，可以容忍5V TTL。

SWIFI 接口：连接SWIFI 核心板。

SWIFI 复位：SW 白色按键，短按改键复位SWIFI。

SWIFI 恢复出厂：此按键用于SWIFI 恢复出厂。短按该按钮，核心板模块红

色指示灯常亮，再短按SW 按钮，模块回复出厂成功。

D2: 电源指示灯。

二、通过网页设置 SWIFI

模块可以通过网页方式进行参数配置。模块出厂默认工作在 AP 模式，此模式下所有支持 802.11 标准的 WiFi 智能终端都可以连接到 SWIFI 模块上，并通过浏览器对模块进行参数配置。网页配置“SWIFI-路由器-云端通信”步骤如下：

- (1) 准备SWIFI 扩展板、核心板、USB 转串口线、电脑一台，如图 1.2 连接，并上电，此时核心板红色指示灯闪烁。

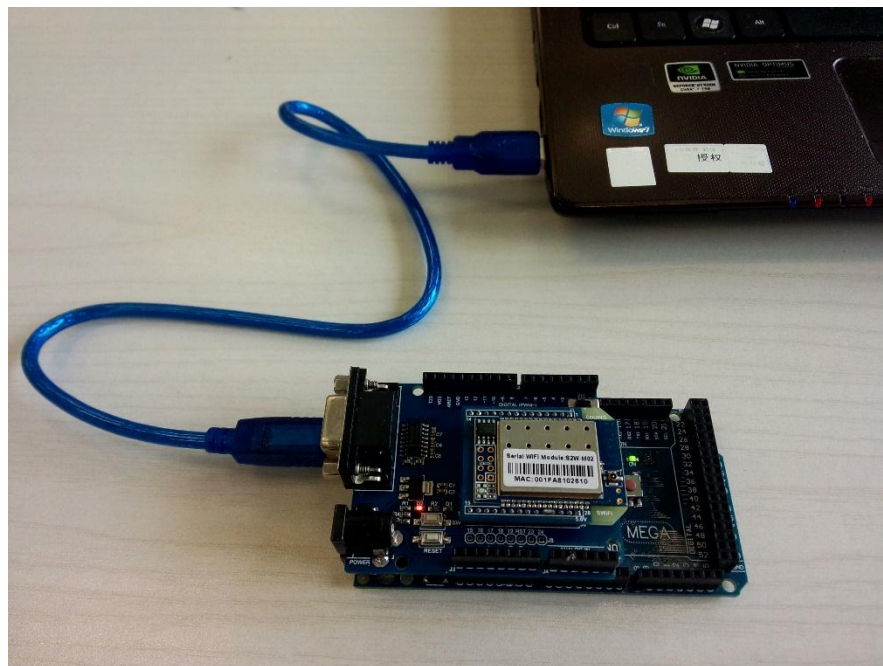


图 1.2 WiFi 模块连接

- (2) 长按 WiFi 模块的 reset 按键(如图 1.3 中绿色方框所示)直至红色指示灯停止闪烁并打开电脑的无线连接模块。连接模块 SSID: SimpleWiFi 或者 SimpleWiFi_xxxx，连接无需密码。

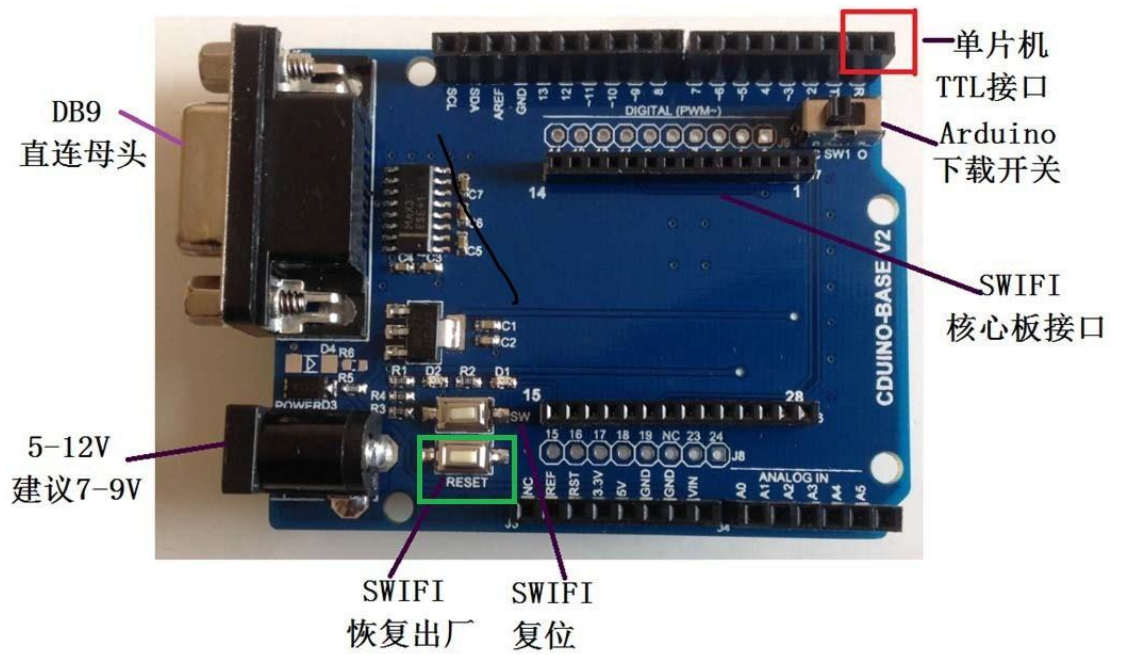


图 1.3 WiFi 复位按键

- (3) 打开电脑的浏览器，输入模块IP 地址：**192.168.2.1**，会出现图 1.4 所示界面(此处为 chrome 浏览器，其他浏览器也可以)：
用户名为 admin，密码：123456。



图 1.4 用户登录界面

- (4) 登录进入后在页面的左上角会有如图 1.5 所示的三个选项：
 BasicSettings, WirelssSettings, AdvancedSettings。

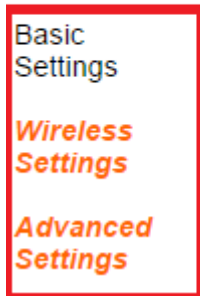


图 1.5 WiFi 设置界面的左上角

(5) 在 Basic Settings 页面进行如下设置:

Settings:

The current settings for Basic Settings may be changed using the form below. To make the new settings apply each time the module is reset. Check the values before pressing the "Submit" button.

	Current	Updated
Baud Rate:	115200bits/second	115200 ▾ bits/S
Data Size:	8bits/character	8 ▾ bits/character
Parity:	None	None ▾
Stop Bits:	1bit(s)	1 ▾ bit(s)
Flow Control:	None	None ▾
Local Port Number:	8000	8000 (range < 200, 65535> or 0 indicate random)
Remote Port Number:	N/A	80
C/S Mode:	Server	Client ▾
Protocol:	TCP	TCP ▾
Server Address:	N/A	202.121.178.242 Domain or IP
Tcp Link Timeout:	120seconds	120 seconds
Retry Connect Time:	N/Amilliseconds	5000 milliseconds
Power Save Mode:	Low	Low ▾
		Submit

先將C/S Mode修改為 Client，再將 Remote Port Number 修改為 80

此處修改為:202.121.178.242

图 1.6 BasicSettings 设置界面

将 C/S Mode 设置为 Client，Remote Port Number 改为 80。Server Address 设置成 202.121.178.242，其他选项不用修改。最后点击 Submit 进行提交。

(6) Wireless Setting 设置界面如下:

SSID 为无线路由名称, NetWork Mode 设置为 infra(STA), Ecrption 设置为 WPA1PSK/WPA2PSK (AUTO), Encryption key 设置为无线路由的密码。设置完成后点击 submit 提交。

Settings:

The current settings for Wireless Settings may be changed using the form below. To make the new settings apply each time the module is reset. Check the values before pressing the "Submit" button.

	Current	Updated
Ssid:	SimpleWiFi_2603	Lab313-9
NetWork Mode:	Infra(AP)	Infra(STA)
Encryption:	Disable	WPA1PSK/WPA2PSK(AUTO)
Key Index:	N/A	1
Encryption Key:	N/A	newbeestudio
Channel:	1	Auto
Wireless Mode:	b/g	b/g
Rate:	Auto	Auto
AP BSSID Filter:	00-00-00-00-00-00	000000000000 (0 indicate Auto)
Wireless Roam:	Disable	Disable

对应路由密码

此处为路由名称

设置和图中相同

Submit

图 1.7 Wireless Setting 界面

(7) Advanced Setting 界面设置如图 1.8 所示:

Address Type 选择为动态 IP (DHCP), 其它选项默认即可。设置完后, 点击 Update Settings 。最后, 点击 reboot 重启 WiFi 模块使设置生效。当 WiFi 模块红色指示灯每次间隔闪烁两下表示连上路由信号。

IP Address Selection

修改如图所示

Address Type:	DHCP/AutoIP
Static IP Address:	192.168.2.1
Subnet Mask:	255.255.255.0
Default Gateway:	192.168.2.1
DNS Server:	192.168.2.1

Update Settings

Port and Password

Port Type:	UART
Data trigger Length:	256 bytes (range <32 , 1024>)
Login Password:	123456 Password must be 6 characters

Update Settings

Restore Factory Defaults

Restore all options to their factory default states: Restore Defaults

图 1.8 Advanced Setting 界面设置

(8) 下载程序测试

Arduino 程序下载时，开关拨至“0”位置，完毕后拨到“C”位置，六件套方可正常工作！

将 SWIFI 模块安装到 Arduino 开发板上，安装如图 1.9 所示。下载程序 wifi.ino 文件至开发板。

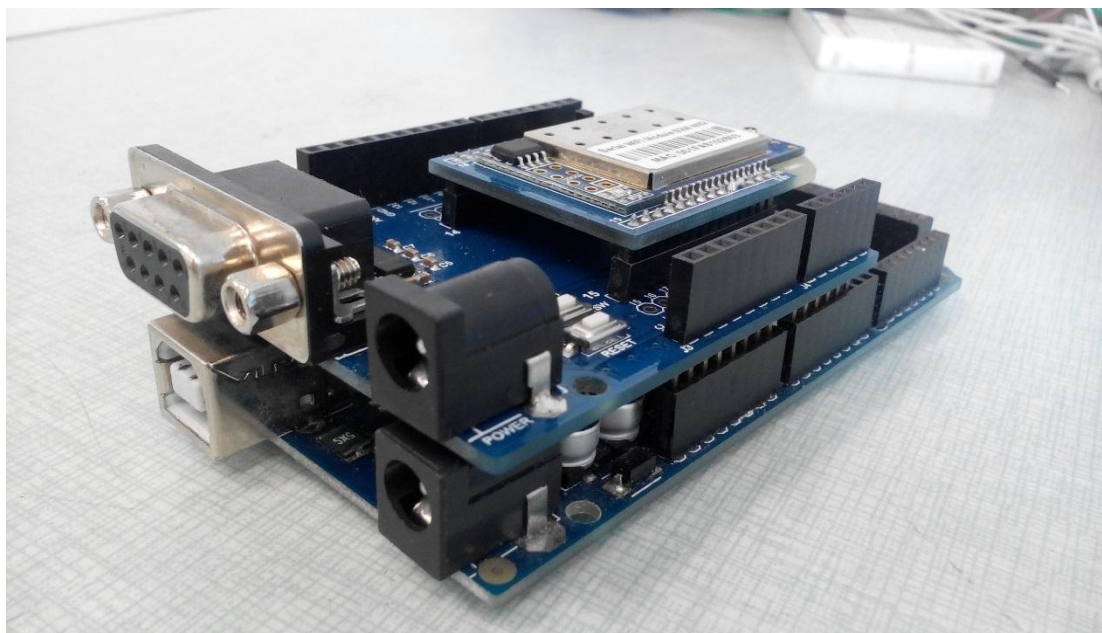


图 1.9 WiFi 模块安装图

(9) 示例程序

- 引用函数库：“aJSON.h”，“DHT22.h”，“Wire.h”，“Adafruit_GFX.h”，“WiFi.h”
- WiFi 对象函数：
WiFi()：构造函数，作用为定义 WiFi 对象。
aJsonObject* PackData(aJsonObject* data, char* resource_id)：公有函数，作用为将数据封装成可发送格式。data 为需要发送的数据，resource_id 为各组自己的数据集 id。该函数返回为 JSON 格式数据。
void PostData(aJsonObject* msg, char* api_key)：公有函数，作用为发送数据。msg 为已经被封装好的数据，api_key 为账户自己的数据 api_key。
- 程序

```
#include <aJSON.h>
#include "DHT22.h"
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <WiFi.h>
#define group_num "0"           //运行时修改为自己的组号
DHT22  DHT(7);                 //定义温湿度传感器
WiFi    wifi;                   //定义 WiFi 对象
```

```

//云端 API KEY, 可以修改为自己的 KEY KEY
char* api = "7b53bb81-195c-477c-8513-b6a7d7bad0d3";
//云端 resource id, 可以修改为自己的 resource id
char* resource = "ee59d976-0a8a-4ee9-b2c3-5f4df614329d";
long previousMillis = 0;
long previousMillis2 = 0;
long previousMillis3 = 0;
int interval1=10;
int interval2=700;
int interval3=1000;
unsigned long currentMillis;
double temperature=0;
double humidity=0;

JsonObject *add_data(float T, float H, char* num)
{
    JsonObject* data = aJson.createArray();
    JsonObject* guid = aJson.createObject();

    aJson.addItemToArray(data, guid);

    aJson.addNumberToObject(guid, "temperature", T);
    aJson.addNumberToObject(guid, "humidity", H);
    aJson.addStringToObject(guid, "group number", num);
    return data;
}
/*****
    *****/
    /*****初始化函数*****/
/*****
void setup() {
    Serial.begin(115200);
}
/*****
    *****/
    /*****主体函数*****/
/*****
void loop() {
    currentMillis = millis();
    //建立串口通信的线程并打印返回的数据
    if(currentMillis - previousMillis > interval1) {
        previousMillis = currentMillis;
        while(Serial.available())
            Serial.write(Serial.read());
    }
    //建立线程更新传感器的数据

```

```

        if(currentMillis - previousMillis2 > interval2){
            previousMillis2 = currentMillis;
            temperature = DHT.getTem();
            humidity =DHT.getHum();
        }
//建立线程发送数据
        if(currentMillis - previousMillis3 > interval3){
            previousMillis3 = currentMillis;
            // 新建 json 数据对象
            aJsonObject* root = aJson.createObject();
            aJsonObject* data = aJson.createArray();

            data = add_data(temperature, humidity, group_num);
            //发送数据
            root = wifi.PackData(data, resource);
            wifi.PostData(root, api);
            //打印调试信息
            Serial.println("*****");
            Serial.println(Serial.read());
            Serial.println("*****");
            //清空缓存，放置数据溢出
            aJson.deleteItem(root);
        }
        delay(1000);
    }
}

```

(10) 数据传输成功效果

数据传输成功，在串口监视器中会有如图 1.10 所示效果。当数据传输成功后，登录网页 <http://data.sjtu.edu.cn>。账户为 hackathon, 密码 111111。进入界面后，点击“我的数据集”，进入下一页面。然后点击 hackathon，预览 group information。这样可以查到上传的数据。

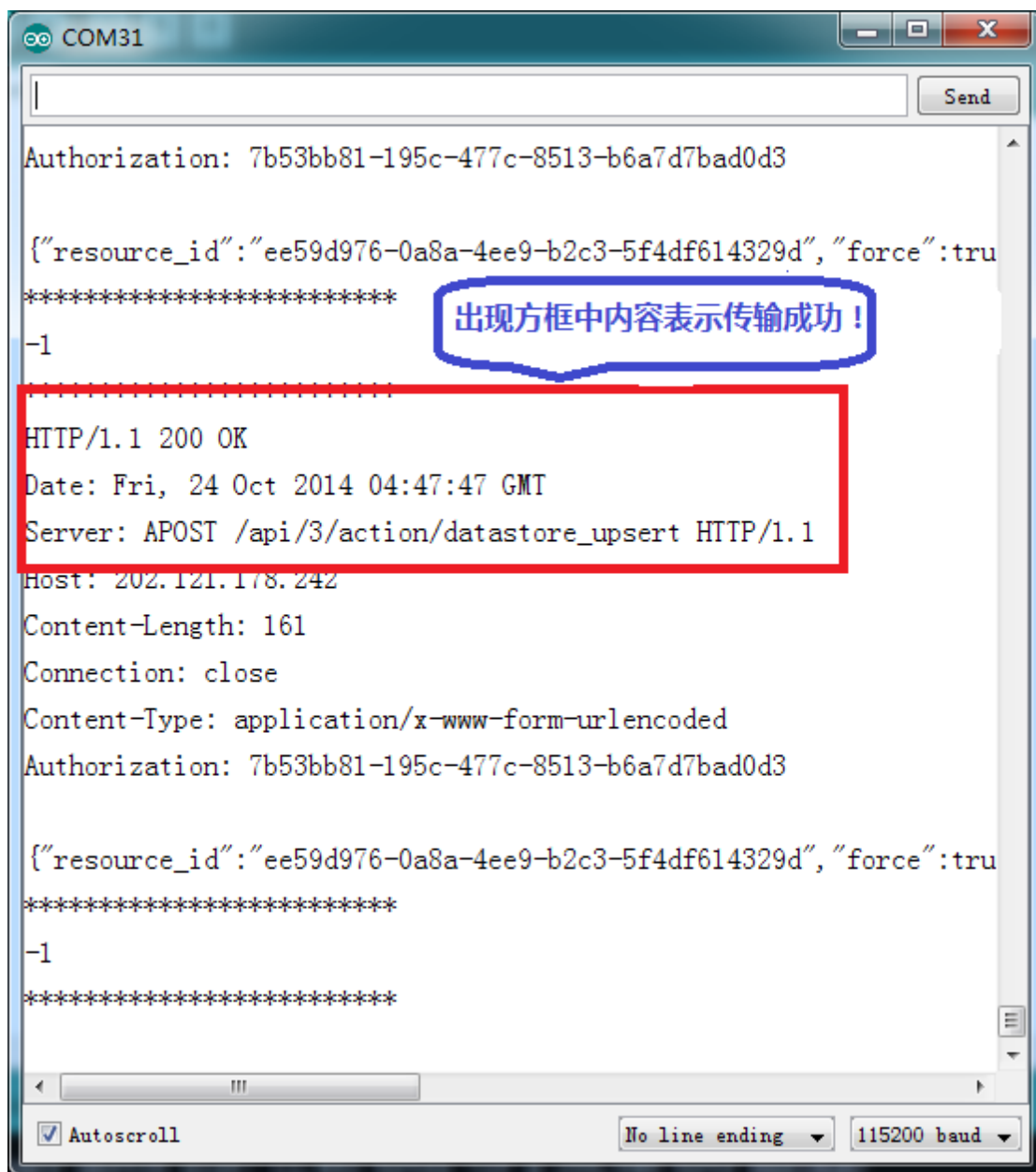


图 1.10 数据上传至云端效果