

# 第一章 获取天气预报(WiFi+https+json+mbedtls)

## 1. 综合演练目标

- 掌握 HTTPS 流程
- 掌握 mbedtls 流程

## 2. 原理

本篇教程所要描述的就是通过免费的天气预报 API 获取对应城市当天以及未来几天的天气情况。通过[按键](#)选择想要[获取天气的城市](#)，然后通过 TCP 连接上免费的天气预报 API，最后一个 GET 请求就可以得到免费天气预报 API 返回的天气信息。

### 2.1. 免费天气预报 API 的选择

现在你在网上搜索关键字"天气预报 API"，就会弹出很多网友分享的接口。但是，据小编东测西踩发现大多都不能用的。有的要么给连接上但是只能获取得到 HTTP 的 Header 内容，然后没有 body 的内容。最终，经过实测小编选择了心知天气的天气预报免费接口；

### 2.2. HTTP VS HTTPS

一般搞嵌入式的工程师大多都只知道加密的区别，由于篇幅原因这里不会贴上一大段关于 HTTP 以及 HTTPS 区别的文章。但是，做为贴心小棉袄的小编已经将通向技术大门的[链接地址](#)附上。[根据这篇链接文章所说的那样，HTTPS 才是以后的方向。](#)

- 因此，此次教程我们就用 HTTPS 的 GET 请求，有别于之前学习的 HTTP 的 GET 请求。

### 2.3. GET 请求

GET 请求只是 HTTP 请求中的其中一种方法，还有另外一个比较常用的是 POST 请求。那么，HTTP 的 GET 请求报文格式是怎么样的呢？

#### 客户端请求消息：

客户端发送一个 HTTP 请求到服务器的请求消息包括以下格式：请求行（request line）、请求头部（header）、空行和请求数据四个部分组成，下图给出了请求报文的一般格式。

请求方法	空格	URL	空格	协议版本	回车符	换行符	请求行
头部字段名	:	值	回车符	换行符	} 请求头部		
...							
头部字段名	:	值	回车符	换行符			
回车符	换行符						请求数据

以下是本次教程的 GET 请求内容：

1	GET /v3/weather/daily.json? key=4nik0ivxfmx fjzz1&location=beijing&language=zh-Hans&unit=c&start=0&days=5 HTTP/1.1 Accept: */* Host: api.seniverse.com
---	---

#### 服务器响应消息

HTTP 响应也由四个部分组成，分别是：状态行、消息报头、空行和响应正文。

```
HTTP/1.1 200 OK
Date: Sat, 31 Dec 2005 23:59:59 GMT
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 122
<html>
<head>
<title>Wrox Homepage</title>
</head>
<body>
<!-- body goes here -->
</body>
</html>
```

状态行

消息报头

空行

下面的就是响应正文了

以下是小编在实战中获取到的状态行+消息报头的内容：

1	HTTP/1.1 200 OK Date: Sat, 30 Jun 2018 08:47:28 GMT Content-Type: application/json; charset=utf-8 Content-Length: 876 Connection: keep-alive Cache-Control: no-cache Pragma: no-cache Expires: -1
---	--

以上部分内容以及更多的 HTTP 详情就参考[菜鸟教程](#)

## 2.4. 加密库选择：乐鑫的库还是第三方的库

当时小编在做出选择的时候挣扎了好久，进行过激烈的思想斗争。最终下定决心选择了第三方的库即 `mbedtls` (更多的详情[请点击我](#))。理由如下：

### ➤ 加解密库

因为在使用 `WiFi` 开发产品的时候，我们会用到各种各样的加解密。有的是有硬件加持，但是更多的还是软件层面上的加解密。正因为这样，`mbedtls` 就更加是不二的选择，因为它有各种各样的如 `base64/sha1/aes128` 等加解密库的同时还支持 `SSL` 以及 `TLS`。更重要的是它背后还有大佬 **arm MBED**

### ➤ 便利性&移植性

专门为了嵌入式而生，即使以后我不再使用乐鑫的 `WiFi` 芯片了，我用了其他的但是又支持 `mbedtls` 的 `WiFi` 芯片，我就可以直接将之前的代码原封不动的复制过去就可以使用了。

## 3. 实战

介绍完原理之后，我想大多数读者已经明白小编是准备使用 `mbedtls` 来实现 `TCP` 连接并发起 `HTTPS` 的 `GET` 请求，最后获取到指定城市的 `json` 格式的天气预报数据内容。那么，要实现这个功能我们需要做些什么呢？

### 3.1. 按键

按键的作用主要是用于选择指定获取哪个城市的天气预报情况，我们可以直接使用[按键单击以及多击的实现](#)中的按键初始化函数即可：`user_app_key_init()`；

### 3.2. RGB 灯

RGB 呼吸灯主要用于显示当前获取的是哪个城市的天气预报, 其中该教程中所对应的映射关系是:

颜色	城市
R	北京&深圳
G	上海
B	广州

直接调用以下的 `rgb` 初始化函数即可: `user_app_rgb_init();`

### 3.3. TCP

这应该是本篇教程最重要的一部分了。但是, 即便是这样我们所要做的事情也不会太多, 大多数要么 `mbedtls` 已经封装好了, 要么就是小编已经封装好了。你只要做的就是填充获取指定城市天气预报的接口即可。

- `https_request_by_GET(HTTPS_URL_BJ);` // HTTPS 的 GET 请求
- `tls_client_handle(GET_REQ);` // TCP 相关的写和读操作

### 3.4. 天气预报接口

正如上面原理中的章节所说, 我们最终选用的是心知天气的接口, 那么我们如何获取到自己的免费接口呢? 登陆[心知天气官网](https://www.xinzhiweather.com/)并注册自己的账号就可以免费获取得到天气预报的接口了, 以下是小编的免费天气预报接口:

1	<code>https://api.seniverse.com/v3/weather/now.json?key=4nik0ivxfmxfjzz1&amp;location=beijing&amp;language=zh-Hans&amp;unit=c</code>
---	--

接口中所表达的详情含义, 这里不再废口舌讲解. 因为心知天气官网[文档](#)已经说得很详细了, 我相信大家都是接受过九年义务教育的大佬, 没有道理看不懂吧。

### 3.5. 天气预报数据解析

心知天气返回的数据是 `json` 格式, 那么我们就需要使用到第三方的开源库 `cJSON` 了。ESP32 的 SDK 已经自带这些移植好的库了, 我们只需要直接使用即可。天气预报的数据格式如下所示:

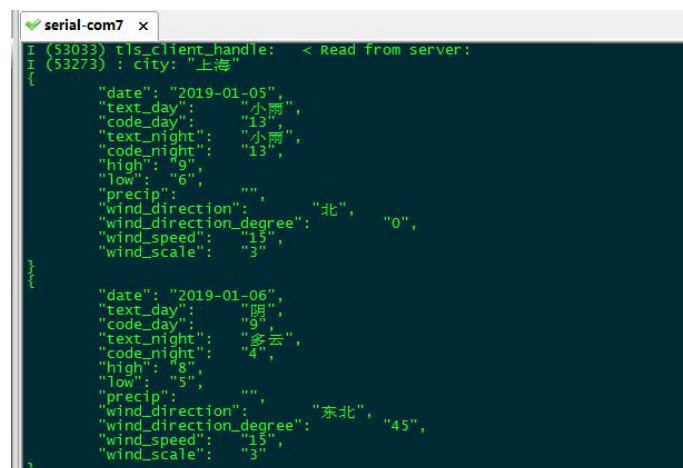
1	{
2	"results": [{
3	"location": {
4	"id": "WX4FBXXFKE4F",
5	"name": "北京",
6	"country": "CN",
7	"path": "北京,北京,中国",
8	"timezone": "Asia/Shanghai",
9	"timezone_offset": "+08:00"
10	},

```

11      "daily": [{                                     //返回指定 days 天数的结果
12          "date": "2015-09-20",                      //日期
13          "text_day": "多云",                        //白天天气现象文字
14          "code_day": "4",                          //白天天气现象代码
15          "text_night": "晴",                        //晚间天气现象文字
16          "code_night": "0",                        //晚间天气现象代码
17          "high": "26",                             //当天最高温度
18          "low": "17",                              //当天最低温度
19          "precip": "0",                             //降水概率, 范围 0~100, 单位百分比
20          "wind_direction": "",                     //风向文字
21          "wind_direction_degree": "255",           //风向角度, 范围 0~360
22          "wind_speed": "9.66",                    //风速, 单位 km/h (当 unit=c 时)、mph (当 unit=f 时)
23          "wind_scale": ""                          //风力等级
24      }, {
25          "date": "2015-09-21",
26          "text_day": "晴",
27          "code_day": "0",
28          "text_night": "晴",
29          "code_night": "0",
30          "high": "27",
31          "low": "17",
32          "precip": "0",
33          "wind_direction": "",
34          "wind_direction_degree": "157",
35          "wind_speed": "17.7",
36          "wind_scale": "3"
37      }, {
38          ...                                         //更多返回结果
39      }],
40      "last_update": "2015-09-20T18:00:00+08:00" //数据更新时间 (该城市的本地时间)
41  }
42  }

```

## 4. 效果展示



```

serial-com7 x
I (53033) tls_client_handle: < Read from server:
I (53273) : city: "上海"
{
  "date": "2019-01-05",
  "text_day": "小雨",
  "code_day": "13",
  "text_night": "小雨",
  "code_night": "13",
  "high": "13",
  "low": "8",
  "precip": "100",
  "wind_direction": "北",
  "wind_direction_degree": "0",
  "wind_speed": "15",
  "wind_scale": "3",
}
{
  "date": "2019-01-06",
  "text_day": "多云",
  "code_day": "9",
  "text_night": "多云",
  "code_night": "4",
  "high": "8",
  "low": "5",
  "precip": "0",
  "wind_direction": "东北",
  "wind_direction_degree": "45",
  "wind_speed": "15",
  "wind_scale": "3",
}

```

```
serial-com7 x
I (108623) tls_client_handle: < Read from server:
I (108703) : city: "广州"
{
  "date": "2019-01-05",
  "text_day": "阴",
  "code_day": "9",
  "text_night": "阴",
  "code_night": "9",
  "high": "18",
  "low": "12",
  "precip": "",
  "wind_direction": "无持续风向",
  "wind_direction_degree": "10",
  "wind_speed": "10",
  "wind_scale": "2"
}
```

```
serial-com7 x
I (140233) tls_client_handle: < Read from server:
I (140293) : city: "深圳"
{
  "date": "2019-01-05",
  "text_day": "多云",
  "code_day": "4",
  "text_night": "阴",
  "code_night": "9",
  "high": "21",
  "low": "15",
  "precip": "",
  "wind_direction": "无持续风向",
  "wind_direction_degree": "10",
  "wind_speed": "10",
  "wind_scale": "2"
}
```

```
serial-com7 x
I (161293) tls_client_handle: < Read from server:
I (161333) : city: "北京"
{
  "date": "2019-01-05",
  "text_day": "多云",
  "code_day": "4",
  "text_night": "多云",
  "code_night": "4",
  "high": "0",
  "low": "-8",
  "precip": "",
  "wind_direction": "北",
  "wind_direction_degree": "0",
  "wind_speed": "10",
  "wind_scale": "2"
}

{
  "date": "2019-01-06",
  "text_day": "多云",
  "code_day": "4",
  "text_night": "多云",
  "code_night": "4",
  "high": "3",
  "low": "-8",
  "precip": "",
  "wind_direction": "西南",
  "wind_direction_degree": "225",
  "wind_speed": "10",
  "wind_scale": "2"
}

{
  "date": "2019-01-07",
  "text_day": "多云",
  "code_day": "4",
  "text_night": "晴",
  "code_night": "1",
  "high": "3",
  "low": "-6",
  "precip": "",
  "wind_direction": "无持续风向",
  "wind_direction_degree": "15",
  "wind_speed": "15",
  "wind_scale": "3"
}
```

## 5. HTTPS 获取天气总结

- 上面有提到 TLS，那么可能很多人就会问了 TLS 和 SSL 到底两者之前有啥区别？它们的工作流程是啥？这里我推荐大家可以看看这几篇文章，写得很不错。
  - [SSL-TLS 双向认证\(一\) -- SSL-TLS 工作原理](#)
  - [SSL-TLS 双向认证\(二\) -- 基于 mosquitto 的 MQTT 双向认证](#)
  - [SSL-TLS 双向认证\(三\) -- ESP8266 与 mosquitto 的 MQTT 双向认证](#)
- 前面 20 章的讲解，大家基本了解 ESP 的网络工作过程了，所以大家[直接怼源码](#)
- 源码地址: <https://github.com/xiaolongba/wireless-tech>