

Anylinkin® ALK8266WIFI®模组

SPI 接口高速通信使用与集成

常见问题 FAQ

目 录

1 前言.....	3
1.1 简介	3
1.2 术语约定	3
1.3 通过书签方便阅读和快速检索定位	4
1.4 参考文档	4
1.5 联系我们	4
2 与 ALK8266WIFI®模组的基础使用有关的常见问题.....	5
2.1 怎么判断 ALK8266WIFI®模组是否正常地启动起来了?	5
2.2 ALK8266WIFI®模组复位的正常时序是怎样的?	5
2.3 ALK8266WIFI®模组上的两个 LED 灯各代表什么意思?	5
3 与 ALK8266WIFI®模组的集成开发使用有关的常见问题	6
3.1 与 ALK8266WIFI®模组的通信初始化过程是怎样的?	6
3.2 主机固件程序里为何要先将 nCS 片选拉低再对模组复位?	7
3.3 如何验证主机固件里 SPI 接口初始化和基本字节读写逻辑是正确的? ...	8
3.4 如何测试验证或获取在可靠通信时主机 SPI 时钟能达到最大频率? ...	8
3.5 SPI 接口片选信号为何要用 GPIO 软件控制?	8
3.6 LED1 (绿灯) 闪烁或灭掉代表什么意思?	9
3.8 模组连接路由器时, 有哪些方法可以查看模组的 IP 地址?	9
4 与网络通信基础有关的常见问题.....	11
4.1 与 AP/STA、TCP/UDP 基本概念有关的基础问题.....	11
4.1.1 AP/STA 服务器/客户端有何区别? AP 可做 TCP 客户端吗?	11
4.1.2 如何用两个 ALK8266WIFI®模组组成一个对等的 UDP 通信? ...	12
4.1.3 为何使用 TCP 通信往往比 UDP 要慢?	12
4.1.4 路由器对 TCP 通信速度为何会有较大的(负面)影响?	13
4.1.5 网络通信时为啥建议一个 TCP 包的大小不要超过 1460?	13
4.1.6 建立 TCP 客户端服务, 为何建议随机产生本地端口而非指定? ..	14
4.1.7 UDP 广播时, 速度为什么很慢不超过 100KBytes/s?	14
4.2 与天线相关的基础常识	14
4.2.1 什么是天线的增益? 为何增益越大, 对方向的对准要求越高? ..	14
4.2.2 为何高增益的天线有可能提高传输速度?	15
4.3 WIFI 的传输距离与那些因素有关? 模组的传输距离有多远?	16
5 与高速通信应用有关的常见问题.....	18
5.1 单片机通过 WIFI 模组上传文件, 对方是一个 FTP 服务器, 需要在单片机上移植一个完整的 FTP 客户端吗?	18
5.2 音视频的录-传-播同步, 需要注意什么?	18
5.3 采用例程测试速度有 1MBytes/s, 但加上高速采集后系统速度降低一半, 这是为什么? 该如何优化?	19
5.4 怎么通过这个模块获取互联网时间?	19
5.5 模组通过路由器联网做 TCP 服务器, 不知其 IP 地址怎么办?	20
6 如有不清之处, 可随时通过 QQ 等手段和我们进行交流	22

1 前言

1.1 简介

Anylinkin® ALK8266WIFI® 是一款价格低廉、灵活、功能强大、高性能、精简小尺寸、绿色环保、高性价比的 802.11 b/g/n 无线模组。它包含有 (1) 高性能且高度集成的无线片上系统芯片 ESP8266EX, 提供智能高效的无线接入; (2) 标准 2.0mm 间距的排针全孔半孔(复合邮票孔), 提供高速通信 SPI 从机接口; (3) 串口数据排针半孔(邮票孔)接口, 提供 UART 串行通信接口; (4) 同时, 还带有一些 IO 外设接口和 LED 灯, 可用于用户扩展。

通过 ALK8266WIFI® 模组所提供的 SPI 半孔整孔复合(邮票孔)接口, MCU 系统 (1) 可以实现和远端 TCP/UDP 服务节点实现高速通信, 最大波特率可达 40Mbps, 实测有效吞吐量可以超过 M 字节每秒, 适用于高速采集、语音、图片以及视频传输等场合; (2) 通过我们提供的基本的 SPI 控制协议, 直接通过 SPI 总线接口就可以对模组及其片上资源进行设置、查询和控制, 无需 UART 串口介入, 以便节约主机的串口线用作其他功能。

本文是在使用 ALK8266WIFI®模组进行单片机 WIFI 通信时的一些常见疑问和回答, 供参考。所牵涉范围不限于模组本身, 也包含许多单片机 WIFI 网络通信上的一些基础实用知识和技巧, 作为《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》辅助参考资料。随着对越来越多用户的支持和探讨的继续, 本文中的常见问题也会不断添加和更新。

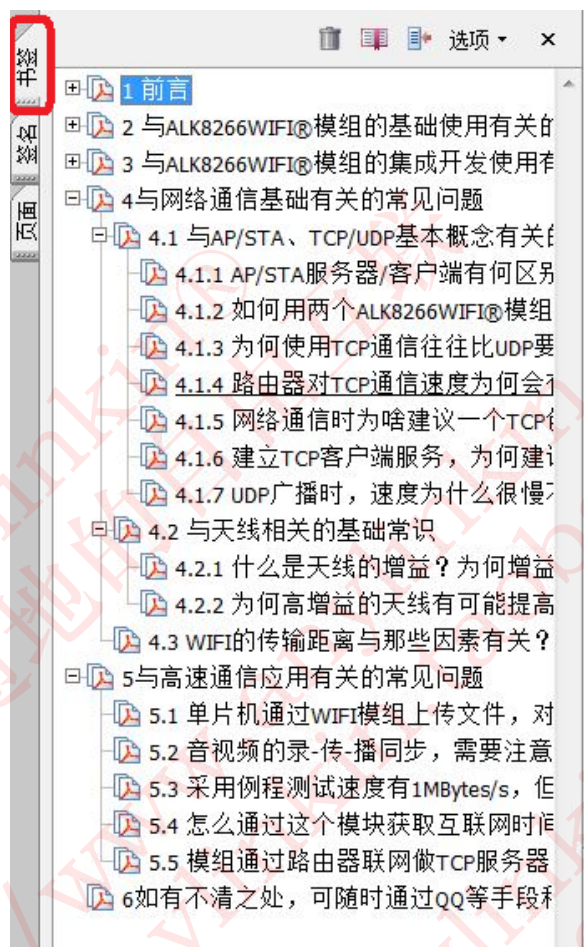
文中如有疏漏或认识不足之处, 欢迎大家积极指出、建议或批评斧正。谢谢!

1.2 术语约定

术语	解释
连接	一个广义的概念, 包括硬件电路上两个网表节点之间的连接、WIFI 工作站(STA)连接热点 AP 或路由器、网络层的 TCP 或 UDP 连接。在本文中, 为了避免混淆, 尽量使用接入和链接, 来区分后两种“连接”。如非特别声明, “连接”一般指硬件上的“逻辑互联”。
逻辑互联	特指硬件电路的两个网表节点之间, 用 PCB 布线、导线飞线等方式连接起来。
接入	特指 WIFI 工作站(STA)使用 SSID 和密码来连接热点 AP 或路由器, 对应英文单词 ACCESS。
链接	特指网络 TCP/IP 层的 TCP 或 UDP 套接字链路连接, 对应于模组上建立的一个服务 (Service)。
服务	一个网络节点开启的功能, 使用 IP 地址和端口标识, 通过套接字将两个节点的服务关联起来, 实现通信, 对应英文单词 Service。例如我们常说的 TCP 服务、UDP 服务等等。

1.3 通过书签方便阅读和快速检索定位

本文档包含相关书签, 请以方便查阅和快速检索, 如下图所示



1.4 参考文档

- 1、ALK8266WIFI 模组数据手册
- 2、ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明
- 3、ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机驱动 API 函数
- 4、常见的 WIFI 模组配网方式简介与对比暨 ANYLINKIN 8266WIFI 模组配网方式和操作说明

1.5 联系我们

网址: <http://www.anylinkin.com>

淘宝: <http://anylinkin.taobao.com>

电邮: IoT@anylinkin.com, 1521340710@qq.com

技术 QQ 群: 148977035, 160356996

如有技术咨询、探讨、或疑惑, 欢迎和我们联系, 提出您的宝贵意见或建议。谢谢支持!

2 与 ALK8266WIFI®模组的基础使用有关的常见问题

2.1 怎么判断 ALK8266WIFI®模组是否正确地启动起来了?

Q: 怎么判断 ALK8266WIFI® 模组是否正确地启动起来了?

A: 要 ALK8266WIFI® 模组能正常使用, 需要模组启动时从板载的 SPI FLASH 加载程序进行启动。正常启动的末期, 模组上的两个灯会走马灯闪烁 2 秒。

如果看到了这两个灯走马灯闪烁, 则表明模组正常地启动起来了, 参看《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》中“4.11.1 启动时两个 LED 灯先交替闪烁”章节的相关说明。

此外, 串口也会输出模组的启动信息, 并最后看到“ready”输出, 这也可以用来判断模组是否正确地启动了。

2.2 ALK8266WIFI®模组复位的正常时序是怎样的?

Q: 为什么我将模组接上单片机但还没有跑程序, 通电后模组依然可能不会启动?

A: 和其他模组比较, ALK8266WIFI® 模组几乎不会遇到启动不起来的情形。但是如果使用方法不对, 也可能会遇到不能正常启动的问题。

当使用 SPI 接口时, 需要确保模组能正常的从 FLASH 启动。可参看《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》中“4.2 模组的硬复位”章节中的相关说明以及本文 [3.2 主机固件程序里为何要先将 nCS 片选拉低再对模组复位?](#)

也可参看我们所提供的《[ESP8266EX 启动的提示信息与上下拉跳线的解释和问题分析](#)》以及《[ESP8266EX 启动信息解释暨刚拿到 ESP8266 模组串口没有输出或输出乱码怎么办](#)》两个文档(位于压缩包“ESP8266 模组基础参考资料_ANYLINKIN.rar”“ESP8266 参考资料_ANYLINKIN\新手入门”目录下), 进行逐一排查。

如果辅助使用我司的 A-LINKER 调试下载工具, 有助于更加便捷定位分析。

2.3 ALK8266WIFI®模组上的两个 LED 灯各代表什么意思?

Q: ALK8266WIFI® 模组上的两个 LED 灯各代表什么意思?

A: 参看《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》中“4.11 模组板载 LED 灯的诊断显示”章节中的说明。

3 与 ALK8266WIFI®模组的集成开发使用有关的常见问题

3.1 与 ALK8266WIFI®模组的通信初始化过程是怎样的?

Q: 对 ALK8266WIFI® 模组的通信初始化过程是怎样的?

A: 步骤如下:

ALK8266WIFI® 模组的通信初始化全部位于 M8266WIFI_ops.c 中的函数:

ALK8266WIFI_Module_Init_Via_SPI()

该函数的核心操作步骤描述如下:

1) 复位模组 -- M8266WIFI_Module_Hardware_Reset()

目的: 复位模组的目的是确保模组有个正确的起点, 包括确保能正确地从板载 FLASH 加载程序等

要求: 必须项

2) 指定模组所使用的 SPI 接口参数 --M8266HostIf_SPI_Select()

目的: 向驱动传输主机所使用的 SPI 和频率

要求: 必须项

3) 配网与连网 – SPI API 函数配网、AT 指令配网、WEB 网页配网、自动配网

目的: 让模块接入路由器

要求: 有条件必须项

如果使用模组的 AP 模式

其他节点来连接这个模组热点所建立的局域网, 那么只需要通过 SPI 接口 API 函数、串口 AT 指令、或 WEB 网页, 来设置该模组的热点 SSID 和密码, 以便其他节点可以接入这个模组 AP 组成的局域网。一旦配置成 AP 模式, 本模块就无须配网。

如果使用模组的 STA 模式

如果以前成功地连接过期望的热点, 且在配网时选择了“保存”, 那么上电复位或掉线后, ALK8266WIFI® 模组会自动连接热点, 所以, 不必重新配网, 这一步可以省略。

否则, 可以通过 SPI 接口 API 函数、串口 AT 指令、或 WEB 网页进行手动配网, 或者自动配网方式(参看我司提供的相关文档《[常见的 WIFI 模组配网方式简介与对比暨 ANYLINKIN 8266WIFI 模组配网方式和操作说明](#)》)。

4) 确保连网成功 -- M8266WIFI_SPI_wait_sta_connecting_to_ap_and_get_ip()

目的: 确保模块联网成功, 以便后续建立 TCP 或 UDP 链接

要求: 有条件必须项

如果使用模组的 AP 模式

不需要本步, 因为工作在 AP 模式下的模组, 使用自身提供的 IP 地址, 一般缺省为 192.168.4.1。

如果使用模组的 STA 模式

需要本步, 以便在后续建立 TCP 或 UDP 链接的之前, 模块已经成功地获取到了 IP 地址。

5) 建立链接 -- M8266WIFI_SPI_Setup_Connection()

完成了上述初始化后, 就可以开始建立服务或与远端节点 TCP/UDP 链接:

M8266WIFI_SPI_Setup_Connection()

然后就可以开始利用套接字链接开始高速收发数据了

对于 TCP 方式的连接, 目前 SPI 高速通信, 支持 ALK8266WIFI® 模组作为 TCP 客户端进行连接远端 TCP 服务器, 也支持 ALK8266WIFI® 模组作为 TCP 服务器。

对于 UDP 方式的连接, 因为是对等连接, 所以无所谓服务器和客户端, 两个节点之间的 UDP 服务可以任意地互联通信。UDP 模式下, 支持单播、组播, 以及广播。

3.2 主机固件程序里为何要先将 nCS 片选拉低再对模组复位?

Q: 在对 ALK8266WIFI® 模组复位时, 固件程序里为什么需要先将 nCS 片选拉低再复位?

A: ESP8266EX 芯片有多种启动模式, 通过启动时芯片的几个 GPIO 管脚上的上下拉电压进行选择。要 ALK8266WIFI® 模组能正常使用, 需要确保模组上的 ESP8266EX 芯片在启动时的模式正确的选择为从 SPI FLASH 启动, 即需要做到 GPIO15 下拉、GPIO0 上拉或悬空、GPIO2 上拉或悬空。

在 ALK8266WIFI® 模组上, GPIO15 下拉已经通过一个 10Kohm 的电阻下拉, GPIO0 通过 100Kohm 以及一个 LED 上拉, GPIO2 悬空。所以, 如果只接电源和地, 上电后, 一般模组就会正常启动起来。

但是, ESP8266EX 芯片的 HSPI 接口对应的 nCS 片选信号是和 GPIO15 复用的, 而大多数的主机主板在进行设计时, 会将 nCS 信号接一个电阻上拉, 所以, 可能会导致模组在上电复位启动时, 从 GPIO15 上读取到的电平为高或不定, 而进入其他的启动模式, 而不是正确地从板上的 SPI FLASH 上启动, 而导致启动失败。

所以, 在上电复位之前, 固件里会有意对模组的 nCS 芯片输出低电平, 以确保 ALK8266WIFI® 模组能正常地从 SPI FLASH 上启动。

3.3 如何验证主机固件里 SPI 接口初始化和基本字节读写逻辑是正确的?

Q: 如何验证主机固件里的 SPI 接口初始化是正确的?

A: 我们提供了一个专用测试函数, 来测试主机固件的 SPI 接口的初始化和基本字节读写是否正确, 以及在某个 SPI 通信频率下, 基本字节的高速读写是否可靠。详情参看《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机驱动 API 函数](#)》中“4.1.2 M8266WIFI_SPI_Interface_Communication_OK”章节。

这个函数会调用 M8266HostIf_Set_SPI_nCS_Pin() 函数来输出片选、并完成几个字节周期的读写。除此之外, 没有任何其他的操作。因此, 可以用来作为最基本的接口初始化和基本字节读写是否正确的判断依据。

验证主机的 SPI 初始化和基本字节读写, 逻辑是正确的, 方法如下:

在按照《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》中的“4.1.2 SPI 主机接口的固件初始化”章节 写好初始化 SPI 接口函数 M8266HostIf_SPI_Init()、nCS 和 nRESET 对应的 GPIO 初始化函数 M8266HostIf_GPIO_CS_RESET_Init()、实现设置复位和片选输出高和低的两个函数 M8266HostIf_Set_nRESET_Pin() 和 M8266HostIf_Set_SPI_nCS_Pin()、基本字节的 SPI 读写函数 M8266HostIf_SPI_ReadWriteByte(), 然后就需要验证这些基础逻辑是否正确。

这个时候, 在执行了上述主机接口的初始化, 并按照《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》中的“4.2.2 模组的硬复位时序的固件实现”方式, 对模组进行复位后, 可以调用单次或单步执行这个函数

ALK8266WIFI_SPI_Interface_Communication_OK() 进行测试。如果这个函数返回正确, 则表明上述初始化和基础函数的实现逻辑、以及连线都是正确的。

为了避免因为时钟过快出现读写字节异常, 在最初进行逻辑测试时, 可以选择较慢的 SPI 时钟, 例如 1-10MHz。

3.4 如何测试验证或获取在可靠通信时主机 SPI 时钟所能达到最大频率?

Q: 如何测试验证或获取主机的 SPI 接口在可靠通信时, SPI 时钟所能达到的最大频率?

A: 参看《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》中的“5.4.5 验证/获取主机板 SPI 接口稳定可靠通信的最大时钟频率”章节。

3.5 SPI 接口片选信号为何要用 GPIO 软件控制?

Q: SPI 接口的片选信号为何要用 GPIO 软件控制? 主机 SPI 片选是封装一体的怎么办?

A: 使用 GPIO 来输出片选(即软件方式控制片选), 是因为在一个片选有效期内, 可以支持多字节连续读和写 SPI 数据。如果采用主机的硬件接口自动配置片选信号输出, 就不够灵活了。所以, 绝大多数单片机主机, 都支持由 GPIO 来软件控制 SPI 片选。

当然, 也可能有极个别非常特殊的单片机主机, 它的片选信号是封装一体的, 或者因为平台对外公开的库做了封装, 导致主机在读写 SPI 时, 只能是读写一个字节, 同时自动

就一个片选周期, 很不灵活(我们一个客户使用第三方平台做二次开发就遇到了这种情形)。

这个时候, 可以采用一种技巧, 同样可以实现片选信号的 GPIO 软件控制, 即, 将主机或库封装一体所对应的片选信号废弃不用(悬空), 不要去接上模组的片选, 重新挑选一个主机 GPIO 作为片选, 并按照《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机驱动 API 函数](#)》中的章节“3.1 M8266HostIf_GPIO_CS_RESET_Init”和章节“3.5 M8266HostIf_Set_SPI_nCS_Pin”实现相关初始化个高低电平输出的实现。这样, 在该单片机上执行 SPI 读写或者调用封装一体的库函数时, 尽管该单片机或封装库, 会在原来封死指定的管脚上输出 SPI 片选, 但是由于该管脚被悬空没有接上模组, 所以, 对模组没有影响, 而真正的模组 SPI 接口的片选, 则由另外指定的这个 GPIO 给出。

3.6 LED1 (绿灯) 闪烁或灭掉代表什么意思?

Q: 接收数据采用查询方式, 单片机大循环固件功能复杂增多膨胀时, 会看到 LED1 等开始闪烁或灭掉, 这代表什么意思?

A: 表明 ALK8266WIFI® 模组接收到了数据, 但是单片机主机没有或者没有及时地读取走数据。此时, 一般需要优化单片机固件大循环中执行时间太长的环节, 避免查询阻塞, 或者改用中断接收方式。参看《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机集成说明](#)》中的“4.9.2 利用数据中断 IO 管脚来辅助诊断和优化单片机主程序的技巧”。

3.8 模组连接路由器时, 有哪些方法可以查看模组的 IP 地址?

Q: 模组连接路由器时, 有哪些方法可以查看模组的 IP 地址?

A: ALK8266WIFI® 模组的功能非常全面和灵活, 有多种方法可以查看模组从路由器上获取到的 IP 地址。

(1) SPI API 函数查询, 如果使用 SPI 接口和模组交互

u8 M8266WIFI_SPI_Get_STA_IP_Addr(char* sta_ip, u16* status);

```
/* *****  
 * M8266WIFI_SPI_Get_STA_IP_Addr  
 * .Description:  
 *   To get ip address of M8266WIFI STA via SPI  
 * .Parameter(s)  
 *   1. sta_ip : the sta ip address returned if successful  
 *               "0.0.0.0" returned if in AP-only mode or ip not achieved  
 *   2. status : pointer to return errcode(LSB) and status(MSB) upon error  
 *               Use NULL if you don't expect them returned  
 * .Return value:  
 *   =1, success  
 *   =0, has error(s)  
 * *****/  
u8 M8266WIFI_SPI_Get_STA_IP_Addr(char* sta_ip, u16* status);
```

(2) AT 指令查询, 如果使用 UART 串口和模组交互

AT+CIFSR\r\n

(3) WEB 网页查询

如果模组的工作模式包含有 AP 模式, 例如 STA+AP 模式, 那么可以通过打开模组上的 WEB 网页, 来查看模组当前所连接的路由器信息, 包括获取到的 IP 地址。

WIFI连接信息

连接状态	CONNECTED
当前SSID	aa7
信道	9
IP地址	192.168.43.89
MAC地址	
信号强度	-37dBm

刷新

连接路由器

SSID	aa7
密码	

☐ 保存
 连接成功!

(4) 路由器直接查看

我们的每个模组都支持路由器测显示对应的 HostName, 每个 HostName 都有 Anylink 的特殊标志, 并且和其序列号存在一一对应关系, 所以便于和路由器上其他接入的节点区分。

- 运行状态
- 设置向导
- 网络参数
- 无线设置
- 带宽控制
- DHCP 服务器
 - DHCP 服务设置
 - 静态地址分配
 - 客户端列表
- 路由功能
- 转发规则
- 家长控制
- 防火墙
- IP与MAC绑定
- 动态DNS
- 系统管理

客户端列表

本页面用于查看当前DHCP服务器的客户端信息。

ID	客户端名	MAC地址	IP地址	有效时间
1				22:22:01
2				19:28:50
3				16:54:12
4				22:09:48
5	Anylinkin! WIFI-001			23:59:28

刷新

这里的显示数值可以通过M8266WIFI_SPI_Set_STA_Hostname()等API函数修改

4 与网络通信基础有关的常见问题

4.1 与 AP/STA、TCP/UDP 基本概念有关的基础问题

4.1.1 AP/STA 服务器/客户端有何区别? AP 可做 TCP 客户端吗?

Q: AP/STA TCP/UDP 服务器和客户端有什么区别? 热点 AP 可以做 TCP 客户端吗?

A: AP (Access Point, 热点) 和 STA (Station, 基站) 是与基础网络的建立有关的一个层次上的概念, TCP 和 UDP 是另外一个层次上与数据包的传输控制有关的同层次概念, 服务器和客户端是建立了 SERVICE 服务之后与 SERVICE 有关的一个同层次概念。

如果要建立 WIFI 局域网, 必须有一个热点 AP (或包含接入功能的路由器), 其他的节点作为 STA 接入这个热点所组成的局域网络。AP 和 STA 都是这个局域网内的节点, 都有唯一的 IP 地址。

TCP 和 UDP 是网络数据传输的控制方式/协议。TCP/UDP 需要先建立相应的服务 SERVICE, 一个服务一般包括 IP 地址和端口。任意的局域网节点, 都可以建立起 TCP 或 UDP 服务, 使用唯一的 (IP 地址, 端口号) 来标识。

SERVICE (服务) 有服务器端 SERVER 和客户端 CLIENT 的概念。但并非所有的 SERVICE 都有 SERVER (服务器端) 和 CLIENT (客户端) 的概念, 例如 UDP 就没有 SERVER (服务器端) 和 CLIENT (客户端) 的概念之分。

TCP 服务必须有一个 TCP SERVER (服务器端), 其他的都是 TCP CLIENT (客户端)。每个 TCP 客户端只能和 TCP SERVER (服务器端) 建立连接和通信, 且 TCP CLIENT (客户端) 在连接 TCP SERVER (服务器) 时, TCP SERVER (服务器) 必须已经建立起来了服务器服务 (SERVER SERVICE), 监控客户端的连接。所以 TCP 连接一般属于非对等连接。TCP 通信底层包含有握手和重发机制。

而 UDP 连接则是对等的, UDP 本身没有 SERVER (服务器端) 和 CLIENT (客户端) 的概念。任何一个节点, 只要在其上启动一个 UDP 服务, 网络内的其他节点就可以向其对应的 (ip 地址, 端口号) 发送数据包。只要发送方在 UDP 数据包中所包含的 (ip 地址, 端口号) 对应上目标节点, 这个节点就可以接收到该数据。UDP 通信底层没有握手和重发机制, 相当于发送方只负责发送出去, 接收方负责根据自己的 (ip 地址, 端口号) 过滤接收数据。

IP 地址可以是单播地址、也可以是组播地址、也可以是 255.255.255.255 这样的广播地址。所以, 当发送一个数据包里包含有广播地址时, 所有的端口号对应上的节点都可以收到数据。只有 UDP 才可以组播或广播。

因此, 完全可以在热点 AP 上建立起一个 TCP 客户端服务。

例如, 让模组 A 工作 AP 模式, 然后让 PC 去连接这个热点获取到 IP 地址。然后在 PC

上建立一个 TCP 服务器, 然后让模组作为 TCP 客户端去连接 PC 上的 TCP 服务器。

4.1.2 如何用两个 ALK8266WIFI®模组组成一个对等的 UDP 通信?

Q: 如何用两个模组组成一个对等的 UDP 通信?

A: 先让这两个模组之间建立网络连接, 然后建立各自的网络服务 SERVICE, 将目标的 IP 地址和端口分别指定为对方的。

第一步, 先建立网络连接。

如果这两个网络节点都通过路由器接入, 那么这两个网络节点本身就是连接的。

如果没有路由器或第三方提供的局域网, 可以将其中一个模组 A 作为 AP 建立一个局域网, 而让另外一个模组 B 来接入它并获取到 IP 地址。作为 AP 的模组 A 本身具有一个 IP 地址, 假设为 192.168.4.1。假设另外一个模组获取到的 IP 地址为 192.168.4.2。

第二步, 建立 UDP 服务连接。

假设准备指定各自的本地端口分别为 1234 和 4321

示意图如下

	模组 A	模组 B
配置的模式	AP	STA
IP 地址	192.168.4.1	192.168.4.2
本地 UDP 端口	1234	4321

MCU 主机配置函数

A: `ALK8266WIFI_Config_Connection_via_SPI(0, 1234, 192.168.4.2, 4321, 0);`

B: `ALK8266WIFI_Config_Connection_via_SPI(0, 1234, 192.168.4.1, 1234, 0);`

函数调用中最后一个 0 表示使用模组的链接号(通道号), 最多有 4 个, 取值为 0~3, 这里举例用 0。

然后在各自的 MCU 主机程序里调用各自的发送和接收函数就可以了。

4.1.3 为何使用 TCP 通信往往比 UDP 要慢?

Q: 在做通信测速时, 同样条件下使用 TCP 通信的速度为什么会比 UDP 慢?

A: 这是因为 TCP 和 UDP 的底层机制有区别。UDP 底层只管发, 没有 ACK 握手和重发机制(参看 [4.1.1 AP/STA 服务器/客户端有何区别? AP 可做 TCP 客户端吗?](#)), 因此其他消耗对带宽的占用少, 所以常常会发现, 使用 UDP 通信的效率比 TCP 高。

此外, 大多数用户在测试 TCP 和 UDP 通信时, 一般都经过了路由器。而路由器对 TCP

的通信速度的(负面)影响比较大(参看 [4.1.4 路由器对 TCP 通信速度为何会有较大的\(负面\)影响?](#)), 所以, 也会导致测试发现 TCP 的通信速度比 UDP 要慢不少

4.1.4 路由器对 TCP 通信速度为何会有较大的(负面)影响?

Q: 在做通信测速时, 即使通信速度远远低于路由器的标称带宽, 但是为什么路由器对 TCP 通信速度, 相对于 UDP 通信, 依然可能会有较大的(负面)影响?

A: 这是因为影响路由器的实际“转发”性能的指标, 除了常见的表征路由器带宽 MBPS 参数之外, 还有一个很重要但是经常被大家忽略的、表征路由器处理能力的指标---MPPS (Mega Packet Per Seconds, 每秒的兆包数), 它表示了路由器每秒能处理转发的 IP 包的数量。

如果将路由器的带宽 MBPS 数值比作邮局每天进出的车辆的运载能力(单位吨), 那么, 包转发率 MPPS 可以类比邮局的分包拣包能力。一般地, 邮局每天处理的包的数量, 受限于人手(包括智能分包拣包), 所以都会有一个最大的限制。而每个包有多重(或多大尺寸), 对每天能处理的包的数量的影响反而不大; 只要每天处理的包的数量 \times 包的重量, 不超过车辆的运载能力就成。假设一个邮局车辆每天的运载能力是 1000 公斤(类比路由器的带宽 MBPS), 每天处理的包的数量最大不能超过 1 万个(类比路由器的包转发率 MPPS), 那么:

- (1) 如果平均每个包裹的重量远远小于 100g (类比网络上存在大量的小包数据), 那么, 车辆每天的运载能力会有富余。对应于在网络传输中, 很难达到路由器的极限带宽, 往往远远小于路由器的极限带宽。
- (2) 如果平均每个包裹的重量都接近 100g (对应于网络上存在大量的大数据包), 那么就会接近车辆的运输能力(逼近路由器的最大带宽)。
- (3) 如果邮局通过内部优化, 提高了分包拣包的能力, 从每天 1 万个包提升到 10 万个包(相当于换了一个包转发率 MPPS 更大的路由器), 那么也有助于发挥路由器的最大带宽性能(即使大量存在小包)。

在通过路由器进行 TCP 通信时, 因为中间存在大量的 ACK 小包, 所以, 当路由器的能力(MPPS)存在瓶颈时(尤其是对于某些便宜的路由器), 这些 ACK 包的存在, 会限制 TCP 通信的实际速度。

4.1.5 网络通信时为啥建议一个 TCP 包的大小不要超过 1460?

Q: 网络通信时为啥建议一个 TCP 包或 UDP 包的大小不要超过 1460 或 1472 字节?

A: 这是因为网络通信上有一个“最大传输单元 MTU”的概念。每次传输的包的大小和 MTU 不相适应时, 可能会带来拆包、粘包(UDP 没有粘包现象)、以及降低通信效率或可靠性等一系列现象。可以 www.bing.com 搜索一下了解更多详情。

尤其是在一些高速实时性通信的场合, 拆包形成的小短包可能造成发送的效率降低; 同时, 因为拆包可能阵发包的数量连贯倍增, 有可能造成接收方来处理不及出现 ACK 丢

弃而造成发送方因为重发次数增多而降低通信效率,甚至可能因为重发次数出现越界而出现套接字状态复位。所以,对于连续高速通信,一般建议 TCP 包长度不要超过 1460 甚至更少比如 1024 以保持一定富裕量,同时也可能与通信系统的窗口长度对齐。

4.1.6 建立 TCP 客户端服务,为何建议随机产生本地端口而非指定?

Q: 为何建议随机产生 TCP 客户端的本地端口?

A: 随机生成的端口,每次会不一样,这样可以避免客户端连续使用相同的 IP 地址和本地端口去链接服务器时,出现被拒绝链接的情形。

TCP 客户端在和服务器建立连接时,会将自己的 IP 地址和端口传递给 TCP 服务器,TCP 服务器会维护这个 IP 地址和端口,作为“存在这个客户端链接”的一种标志。而一旦客户端异常断开链接,例如调试过程中的直接复位,这种客户端侧的异常断开链接,可能并不会通知服务器,于是服务器将继续“维持该链接的记录”,直到超时后删除。

如果客户端在这个超时时间段重新连接该服务器,并使用了和上一次连接相同的本地端口,那么在服务器那边,就会出现和之前依然存在的 TCP 客户端链接完全一样的信息,绝大多数服务器的处理方式,会将其当成冲突而拒绝这个新的链接请求。在客户端这边就表现为,异常断开后再次重连服务器,会出现失败,而需要等待较长时间后,才可以继续使用这个端口再次去链接服务器。

为了避免出现这种情形,建议本地的端口,由客户端自行随机产生,随机产生的端口往往每次会不一样,所以可以避免出现被服务器拒绝链接的情形。

4.1.7 UDP 广播时,速度为什么很慢不超过 100KBytes/s?

Q: UDP 单播测试速度有 1Mbytes/s,但是改成广播后,速度立刻降低到不到 100KBytes/s,这是为什么?

A: 为了兼容各种不同的节点设备,在 802.11 无线系统上,广播帧(甚至也包括组播帧、以及一些管理帧),都会以缺省最低的速率传播,这个数值一般是 1Mbps,大约相当于有效速度不到 100KBytes/s。当然,在某些系统上,这个缺省的最小值近来有所提高,但是基本上都会为了兼容各种不同的节点设备,广播的速度依然会采用一个较小的速率,而非适配器本身所支持的速率。

4.2 与天线相关的基础常识

4.2.1 什么是天线的增益?为何增益越大,对方向的对准要求越高?

Q: 天线增益是个什么概念?为何天线的增益越大,对方向的对准要求越高?

A: 天线本质上是一个将电磁波对外辐射的无源器件。如果天线只是空间里的一个孤立点,不难理解,其辐射的模式会是一个球体,均匀地向四周辐射,每个球面上的点所接收到

的信号强度是一样的。而天线增益的本质,是将向空间球面辐射的信号,集中压缩到某个方向(平面上、圆柱圆锥、或甚至直线上),从而增强某个接收方向上的信号强度。这种增强后的信号强度,与没有增强之前自然球面辐射所得到的信号强度之比,就是这个天线的增益。所以,这种增益,是以牺牲其他方向上的信号传播为代价,来增强有用/期望的方向上的信号。

打个比方,当我们用手做出喇叭状,对着我们想说话的人时,那个人听到的声音将会增强,但是其他方向上的人听到的声音就会减小甚至可能听不见了。

因为天线本质上是将其他方向上的信号辐射,压缩集中到某个方向上,且增益越高的天线,一般压缩的集中度越高,方向方位越窄,因此对于方向的对准要求或难度也就越高,也可能会相应增加部署的难度。

购买天线时,尤其是高增益天线,一般需要参考其方向图,找到其对应的最大方向,确保通信方处于这个方向上。否则,天线可能反而起到反效果。

4.2.2 为何高增益的天线有可能提高传输速度?

Q: 为何高增益的天线有助于提高传输速度(即使不接天线时的信号强度已经够强了)?

A: 和有线通信相比,无线通信更容易受到周围环境的同频干扰,接收方所处的环境,不仅有来自所期望的发送方的信号,同时,在空间还存在着无数的其他同频信号。这些同等强度甚至更高强度的信号的干扰,会导致接收方的接收效率降低,从而表现出来的系统速度降低。

当发送方发送过来的信号以足够的强度到达接收方(接收方足够感知得到)时,接收方不仅感受到这个所期望的信号,也会同时感知到周围空间存在的其他信号。而接收方是否可以成功接收,还取决于接收方对这些信号的处理/过滤能力,或者说抗干扰的能力。对其他信号的过滤处理,也会影响有用双方的通信效率/速度。

这个场景,可以类比于两个人相距一段距离互相说话。如果周围没有其他人说话,那么他们的交流会比较清晰顺畅。但是如果周围有其他人用相似的声音强度同时说话,就会对这两个人的交流产生干扰,导致一方无法听清楚或听完整另外一方的讲话,结果就会是,听话的一方只能听到部分而丢失了部分讲话,或者说话的一方不得不多次重述,于是交流的效率(速度)就降低了。即使原本两个人交流的声音足够强,只要和周围其他人说话的声音没有形成明显的落差,依然也免不了出现这种情形。

如果此时这两人,说话的一方用手做出一个喇叭状对准听话的一方,听话的一方会感觉到其声音明显压过周围其他人的声音,此时交流又会回归清晰顺畅,这是因为增强声音后,听话的一方就很容易将说话人所说的话,从其他周围的声音中区分开来。

在无线通信的场合,一般大量存在着许多足够强度同频信号。外接增益天线相当于就起到了上述类比中喇叭的作用,从而让接收方所接收到的信号强度明显超过周围的其他信号,减少接收方处理效率降低、接收丢包等因素导致的发送方不得重发等现象所带来的(表现)速度降低的情形。

从上面的描述也可以看出,增益天线对速度的贡献,有一个上限,这个上限是通信系统

固有的最高速度。到了这个速度之后,即使提高天线增益,也不会再提高通信系统的速度。所以,外加天线增益只需要确保到达接收方的信号强度和周围的噪声有显著落差就可以了,没有必要过分强调增益,在周围很干净没有其他同频噪声干扰的场合,增益天线对速度的正面作用可能也不大。

4.3 WIFI 的传输距离与那些因素有关? 模组的传输距离有多远?

Q: WIFI 的传输距离与那些因素有关? 模组的传输距离有多远?

A: WIFI 的可接收的传输距离大体上主要受以下几个要素的影响:

- 1) 发射方达到接收方时的信号水平强度,以及接收方的信号敏感度,以及
- 2) 系统所期望或接受的速度要求

等因素所决定。

1 发射方达到接收方时的信号强度

(或者,更严谨地说,是发送方的信号作为有用信号,相对于周围信号作为噪声,所得到的“信噪比”,能超过接收方的感知下限)

理解这个概念,可以参照我们经常在电脑上扫描某个信号强度对应的信号数值。

这个要素,往往包括以下几点:

(1) 发射方的发射功率。

发射功率越大,传输到接收方的信号自然越强,或者说,在保持同样接收信号强度的前提下,传输的距离可以更远。不过发射功率越大,意味着系统的功耗也将越大。

(2) 发射方天线的增益

参考 [4.6.1 天线增益是个什么概念? 为何天线的增益越大,对方向的对准要求越高?](#), 天线增益有助于将信号压缩集中到传输放上,当然也就可以提高到达接收方的信号强度。

(3) 周围环境的衰减

例如墙壁或者潮湿的空气等等,都可能会对信号的传输形成显著的衰减。

上述各个因素都不是完全由 WIFI 模组唯一决定的。

2 系统所期望或接受的速度要求

当到达接收方的速度较弱时,系统的传输速度会因此而降低([4.6.2 为何高增益的天线有助于提高传输速度\(即使不接天线时的信号强度已经够强\)?](#)),而较远的传输距离,往往意味着到达接收方的信号会减弱。

因此,如果对速度的要求不是很严格,那么传输距离可以适当拉长。而如果希望传输速

度接近通信系统的最大效能, 则传输距离不宜较远。

所以, 笼统地说 WIFI 模块的传输距离有多大, 其实是不严谨的, 尽管基本上说一个大致的范围并没有错。

3 本模组的传输距离

本模组实测结果显示, 在不外接天线时, 普通室外晴天环境, 视线距离 100 米左右, 依然可以保证速度不低于近距离速度的 70% (例如, 近距离测试时, 如果速度为 1M 字节每秒, 视线距离 100 米左右时, 速度不会低于 700K 字节每秒)。而在外接增益天线时, 强增益天线可以确保 1KM 以上的传输距离, 而视线传输距离在 500 米以上, 速度可以确保速度依然维持在近距离速度的 70% 以上。

5 与高速通信应用有关的常见问题

5.1 单片机通过 WIFI 模组上传文件, 对方是一个 FTP 服务器, 需要在单片机上移植一个完整的 FTP 客户端吗?

Q: 单片机通过 WIFI 模组上传文件, 对方是一个 FTP 服务器, 需要在单片机上集成一个完整的 FTP 客户端吗?

A: 没必要。

单片机是一个小型的智能系统, 其优势在于精简, 如果仅仅为了传输一个文件, 在单片机上集成一个完整的 FTP 客户端, 不仅浪费资源, 降低效率, 而且可能因为单片资源不够而无法实现。

所以, 我们完全可以只抽取其中所需要的部分功能, 实现和 FTP 服务器的正常通信和文件上传。具体做法, 我们称之为“模拟复制”实现方法, 具体说明如下:

- 1、找一个 FTP 客户端工具, 执行操作, 向 FTP 服务器上传一个文件, 不需要有其他的操作。

- 2、在 1 中操作的同时, 打开 Wireshark 工具, 对此过程进行抓包。

这个工具很强大, 还会帮助你分析各个包, 所以, 很容易看到 TCP 包的内容。

- 3、将 2 中得到的 TCP 包的内容, 在 TCP-USR-232 等工具里, 以 TCP 包数据的形式直接发送, 观察服务器端的反应, 进行验证。

- 4、将 3 中验证正确的 TCP 包数据, 转移到单片机中, 作为 TCP 包向 FTP 服务器发送并处理服务器的相关的必要反馈, 即实现在单片机上的简单的 FTP 文件上传。

5.2 音视频的录-传-播同步, 需要注意什么?

Q: 在单片机上测试 WIFI 模组的通信速度能到 1M 字节每秒, 而且不丢数据, 但是为什么用来传输视频播放时(需要的速度只有 500K 字节每秒)时, 依然出现视频卡顿的情形? 这个问题该如何解决?

A: 这是因为, 受网络环境因素等各种因素的影响, 网络通信的速度一般都是波动的, 很难保持恒速, 因为网络通信是一种流控制, 而非实时传输。但是视频的录或播, 往往是时间均匀的, 所以存在一种可能性, 就是在某个时刻, 网络的传输速度低于了视频播放所需要的速度, 而可能因为播放器没有数据输入播放而出现了卡顿。

恰当地使用先进先出缓存, 则可以很好地解决音视频录传播的同步问题。

缓存的作用可以理解成, 当网络的速度超过播放的速度时, 可以先预存一部分先进先出的数据, 以便在网络速度偶尔低于播放速度时, 可以有这部分被缓存的数据接力播放。

这个缓存的大小, 需要根据实际的情形做优化处理, 如果网络速度波动容易加大较长, 则缓存可能会需要较多。

我们看有一种体验, 许多在线播放器, 一般都一个缓冲区参数设置。当我们将这个参数设置得太小, 播放时就会经常出现卡顿, 即使我们的网速够快。但是一旦我们将这个缓存设计得合适了, 播放会非常流畅。而如果将这个参数设置得太大, 也是一种浪费。此外, 我们也会看到, 网络传输并非一直都在进行, 而是会根据缓存里的数量大小, 时而启动, 时而停止。这些, 都可以用来作为单片机做音视频同步录传播的思路参考。

所以, 做实时录传播, 不只是 WIFI 模组的速度越快越好, 而只要 WIFI 模组的平均速度能超过录和播的速度就可以了, 重要的是, 处理好先进先出的缓存机制。

5.3 采用例程测试速度有 1MBytes/s, 但加上高速采集后系统速度降低一半, 这是为什么? 该如何优化?

Q: 执行例程测试速度稳定在 1Mbytes/s, 加上高速采集模块代码后, 速度降低到 450KBytes/s, 这是为什么? 该如何优化? 采样速度为 800KB/s。

A: 这可能是因为单片机串行而非并行执行的结果, 导致“系统”的速度降低。

假设每次采集 800 个字节就发送一次, 那么, 在串行执行模式下, 以 800KBytes/s 采集速度, 采集 800 个字节需要占用 CPU 平均约 1ms 的时间, 然后以 1MBytes/s 的速度通过 WIFI 发出去, 则需要平均约 0.8ms。所以, 从整个系统的角度来看, 完成 800 个字节的采集和传输, 占用了 1.8ms 的时间, 和系统平均速度为 $800/1.8=444\text{KBytes/s}$ 。

在这里可以看到, 系统的速度不仅远远低于模块的及时通信速度 1Mbytes/s, 也远远低于高速采集的速度。这是因为, 串行执行的结果导致模块和采集器都未发挥其最大性能。

采用并行技术, 辅以 FIFO 缓存, 例如, 将高速采集改成 DMA 而非由单片机来执行处理, 相当于有“2 个 CPU”在流水线并行处理, 可以大大提高系统整体的吞吐速度, 最终的结果可以让系统的吞吐速度逼近 800KBytes/即采集的速度。

5.4 怎么通过这个模块获取互联网时间?

Q: 怎么通过这个模块获取互联网时间?

A: 通过 WIFI 模块获取互联网时间非常简单, 只要让模块接入互联网, 作为客户端去访问某个互联网服务器就成。

步骤 1: 确保模块接入了互联网, 比如, 链接 1 个可以接入互联网的路由器

步骤 2: 建议一个 TCP 链接, 模块作为 TCP 客户端, TCP 服务器的参数如下:

目标地址: www.beijing-time.org

目标端口: 80

步骤 3: 向这个服务器发送 GET 请求, 获取时间, 即将如下字符串作为 TCP 包的数据发送出去:

"GET /time15.asp HTTP/1.1\r\nHost: www.beijing-time.org\r\n\r\n"

步骤 4: 从所接收到的数据中, 解析出互联网时间来。

接收到的数据样例如下 (或许会有细微的差别), 末尾就是互联网时间

HTTP/1.1 200 OK

Cache-Control: private

Content-Length: 97

Content-Type: text/html

Server: Microsoft-IIS/7.0

Date: Sun, 08 Apr 2018 10:56:36 GMT

t0=new Date().getTime();

nyear=2018;

nmonth=4;

nday=8;

nwday=7;

nhrs=18;

nmin=56;

nsec=37;

5.5 模组通过路由器联网做 TCP 服务器, 不知其 IP 地址怎么办?

Q: 模组通过路由器联网, 每次都是 DHCP 获取得到动态地址。所以客户端去链接它时如何知悉其 IP 地址?

A: 有两种方法可以解决这个问题: 方法一, 对 TCP 服务器模组设定静态 IP 地址; 方法二, 利用 UDP 的广播属性获取做 TCP 服务器的模组的 IP 地址。

方法一, 静态 IP 地址:

可以调用 API 函数 M8266WIFI_SPI_Config_STA_Static_IP_Addr() 来对模组预设静态 IP 地址。

详情可参看《[ALK8266WIFI 模组 SPI 接口高速通信使用与集成—主机驱动 API 函数](#)》中的章节“4.2.4 M8266WIFI_SPI_Config_STA_Static_IP_Addr”

方法二, 利用 UDP 的广播属性获取做 TCP 服务器的模组的 IP 地址

我们知道, UDP 相较于 TCP 的一个优势是, 支持广播帧 (目标地址为广播地址), 广播帧可以被网段内的所有节点接收到, 然后比较目标端口是否一致决定是否接受处理。我们可以利用这个特点来提取做 TCP 服务器的模块的 IP 地址和端口信息。

同时, ALK8266WIFI® 模组支持多链接, 可以方便的配合实现对 TCP 服务器的 IP 地址的提取。

具体说明如下(为了方便说明, 我们以下称做 TCP 服务器的模组为节点 A, 而准备作为 TCP 客户端需要去链接 TCP 服务器的节点--可能是手机、电脑或者另外一个模组--称为节点 B):

- (1) 节点 A 上电后, 除了建立一个 TCP 服务器服务外, 同时也建立一个 UDP 服务, UDP 服务的本地端口预先指定已知。
- (2) 节点 B 在连接节点 A 上的 TCP 服务器之前, 先建立一个 UDP 服务, 来发送一个 UDP 广播包。这个广播包的目标地址为广播地址, 例如 255.255.255.255, 端口为节点 A 预先设定的 UDP 本地端口, 广播包的内容可以是自己定制的数据协议格式, 表示“询问某某 TCP 服务器的 IP 地址”。
- (3) 因为节点 B 发送的 UDP 广播包的目标端口是节点 A 上 UDP 服务的端口, 而目标地址是广播地址, 节点 A 的 UDP 服务会接收到这个包, 解析后知道是节点 A 在询问自己的 IP 地址。在接收这个 UDP 的同时, 节点 A 也知道了节点 B 的 UDP 服务的 IP 地址和 UDP 本地端口。
于是节点 A 可以以节点 B 的 IP 地址和 UDP 本地端口为自己的目标地址与目标端口, 返回一个 UDP 包(此时就不需要是广播包了), 包的内容包含了自己的 IP 地址。
- (4) 于是节点 B 就知道了节点 A 的 IP 地址了。于是就可以继续作为 TCP 客户端建立对节点 A 的连接了。

对方法二的补充说明:

- (1) 上述方法不仅仅可以传递 TCP 服务器的 IP 地址, 也可以用来传递 TCP 服务器的端口等其他信息。只要定制好(2)(3)所有提到的数据协议格式就成。
- (2) 上述方法也可以用于接入认证等功能实现。比如在方法二的(3)中, 节点 A 同时会对数据的有效性进行分析, 合法的才返回自己的 IP 地址和 TCP 服务器的端口, 才允许被连接。

比较方法一和方法二:

- (1) 方法一, 直接设置静态 IP 地址, 实现简单。

但是可能存在地址冲突的现象, 即指定的静态 IP 地址可能已经被本网段其他的节点给占用了。此时可以在路由器里绑定 IP 地址(需要现场的路由器支持这么做, 且不会给运营带来麻烦)。

- (2) 方法二, 实现灵活, 适应面广, 推荐使用这种方法。

但是实现起来可能不如方法一简单直接。其实也不难, 只是比方法一稍微多几个步骤。

6 如有不清之处, 可随时通过 **QQ** 等手段和我们进行交流

Anylinkin®
随时随地的自由互联
<http://www.anylinkin.com>
<http://anylinkin.taobao.com>
IoT@anylinkin.com
1521340710@qq.com

非公开声明

本文档及其相关资料(包括相关文档和参考例程等),仅授权购买我司高速 WIFI 模组 ALK8266WIFI® 的客户(公司)使用参考,其他人员不得使用或参考。

凡是经过我司正常渠道接收本文档及其相关资料的用户,即获得使用本文档及其相关资料的授权。

未经我司同意,授权客户不得对外公开、分享,或转让本文档及其相关资料的部分或全部。