

# WIFI

---

[Wifi协议\(基础篇\) - 知乎\(zhihu.com\)](#)

- **AP(Access Point)**: 无线接入点, 这个概念特别广, 在这里, 用大白话说, 你可以把CC3200当做一个无线路由器, 这个路由器的特点不能插入网线, 没有接入Internet, 只能等待其他设备的链接, 并且智能接入一个设备。
- **STA(Station)**: 任何一个接入无线AP的设备都可以称为一个站点。大白话说也就是平时接入路由器的设备
- **SSID(Service Set Identifier)**: SSID,每个无线AP都应该有一个标示用于用户识别, SSID就是这个用于用户识别的名字, 也就是我们经常说到的wifi名。
- **BSSID**: 每一个网络设备都有其用于识别的物理地址, 这个东西呢就叫MAC地址, 这个东西一般情况下出厂会有一个默认值, 可更改, 也有其固定的命名格式, 也是设备识别的标识符。

## 在无线环境中STA接入的过程包括:

---

认证STA有没有权限和接入点(AP, AccessPoint)建立链路;  
STA能不能接入WLAN;  
以及STA接入WLAN网络之后, 认证STA能不能访问网络的权限

在STA和AP建立链路的过程中, 当STA通过信标(Beacon)帧或探测响应(Proberesponse)帧扫描到可接入的服务集标识符(SSID, ServiceSetIdentifier)后, 会根据已接收到的Beacon帧Proberesponse帧的信号强度指示(RSSI, ReceivedSignalStrengthIndication)来选择合适的SSID进行接入。

# Wi-Fi网络主要由以下三种帧类型构成：管理帧、控制帧和数据帧。

---

WLAN数据帧的基本格式：

1. 帧控制字段：描述帧类型、子类型和控制信息。
2. 管理帧和控制帧的额外字段：这些字段包括帧的持续时间、接收端MAC地址、发送端MAC地址、目标BSSID地址等。
3. 数据帧的额外字段：这些字段描述了上层协议和数据的类型和长度。
4. 数据载荷：包含传输的数据。

## 蓝牙

---

[蓝牙：蓝牙协议-CSDN博客](#)

蓝牙是一种近距离的无线传输技术，由爱立信公司在1994年创立，现在由蓝牙技术联盟管理和更新，

- 蓝牙用于在不同设备之间建立连接，
- 蓝牙的工作方式跟WiFi非常类似它利用无线电波在短距离设备之间发送数据，
- 不过不同的是WiFi信号在路由器与设备之间传输数据，而蓝牙是在设备与设备之间传输数据，这种传输以千兆赫兹(GHz)的频率进行，
- 通常WiFi和蓝牙在2.4GHz的频率下工作，所以蓝牙响应速度非常快，

尽管蓝牙与WiFi使用同频率工作，但是蓝牙信号要比WiFi信号弱的多，它的功率仅为 一毫瓦(mw)，

所以传输距离也有限最初的蓝牙1.0，传输距离只有10米，

发展到现在的蓝牙5.0，传输距离可以达到300米

一般一个蓝牙主设备可以连接七个子设备，但同样的设备只能同时

连接一个，比如不能同时连接两个蓝牙耳机，

因此蓝牙键盘鼠标，耳机等不同设备可以同时连接主设备因为蓝牙收发器在工作频率上有79的通道，也就是79个频率

而且通道的切换速度为1600次每秒，所以不用担心子设备之间互相影响

两台设备连接时，

两者会进行一次配对，他们互换指令并决定是否需要交换数据,或者一台设备是否需要控制另外一台

在结束配对后，两台设备确定好彼此的角色，它们相互连接并形成  
一个网络，这个网络被称为微微网

### [STM32F103蓝牙串口模块HC05编程实验哔哩哔哩bilibili](#)

```
while(HC05_Init()) //初始化ATK-HC05模块
{
    Serial_Printf("HC05_Init error\r\n");
    Delay_ms(500);
}
```

```
HC05_Role_Show();
```

```
//显示ATK-HC05模块的工作状态
void HC05_Role_Show(void)
{
    if(HC05_Get_Role()==1)
    {
        Serial_Printf("Master\r\n"); //主机
    }
    else
    {
        Serial_Printf("Slave\r\n"); //从机
    }
}
```

```

RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE); //使能GPIOA外设时钟

GPIO_InitTypeDef GPIO_InitStructure;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4; // 端口配置
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU; //上拉输入
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度为50MHz
GPIO_Init(GPIOA, &GPIO_InitStructure); //根据设定参数初始化A4

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5; // 端口配置
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度为50MHz
GPIO_Init(GPIOA, &GPIO_InitStructure); //根据设定参数初始化GPIOA5
GPIO_SetBits(GPIOA, GPIO_Pin_5);

USART2_Init(9600); //初始化串口2为: 9600, 波特率.

while(retry--)
{
    HC05_KEY=1; //KEY置高, 进入AT模式
    Delay_ms(10);
    u2_printf("AT\r\n"); //发送AT测试指令
    HC05_KEY=0; //KEY拉低, 退出AT模式
    for(t=0; t<10; t++) //最长等待50ms, 来接收HC05模块的回应
    {
        //USART2_RX_STA&NV99999\

```

```

HC05_Role_Show();
//HC05_Sta_Show();
while(1)
{
    Delay_ms(10);
    if (strcmp((char *)USART2_RX_BUF, "SMO") == 0)
    {
        sendmask = 1;
    }
    else if (strcmp((char *)USART2_RX_BUF, "SMC") == 0)
    {
        sendmask = 0;
    }
    if(t==50)
    {
        if(sendmask) //定时发送
        {
            Serial_Printf("HC05 %d\r\n", sendcnt);
            u2_printf("HC05 %d\r\n", sendcnt); //发送到蓝牙模块
            sendcnt++;
            if(sendcnt>99)
            {
                sendcnt=0;
            }
        }
        //HC05_Sta_Show();
        t=0;
    }
}

```

## GPS

[开源GPS北斗小型追踪器, 汽车轨迹跟踪器, 合宙lua开发的NB-IOT, 北斗定位, MQTT通讯, 全栈开发哔哩哔哩bilibili](#)

[p7\\_stm32从GPS模块获取标准时间 算法哔哩哔哩 bilibili](#)

通过GPS模块获取信息: 时间+经纬度。设计一个中断函数, 解析GPS模块发过来的数据: 数据里时间字符比较混乱, 还需要转换为北京时间 (+8小时)

\$GNRMC, 005244.693, V, , , , , , 110620, , , N\*5C  
\$GNVTG, , , , , , , , N\*2E  
\$GNZDA, 005244.693, 11, 06, 2020, 00, 00\*45  
\$GPTXT, 01, 01, 01, ANTENNA OK\*35

\$GNGGA, 140141.000, 2267.69320, N, 11319.03536, E, 1, 20, 0.7, 44.4, M, -6.7, M, , \*6A