1. C++
2. MCU/Rots/ QNX
3. I2C、SPI、UART、SDIO、USB等电气特性
4. TCP/IP，HTTP，HTTPS等网络协议. 包括包过滤、DHCP/HTTP等常见网络协议，对VLAN、QoS、MAC等二层交换技术了解
5. 短距离通信开发方向(BT/GPS/NFC/WIFI)
6. 蓝牙、Wi-Fi等无线通讯协议开发(ffmpeg，gstreamer、OpenCV)
7. x86/ARM等主流的系统架构
8. 单片机和一些模块

**89C51**

学习目标：proteus仿真、Keil4和Keil5、ISIS、线路搭建、开发板烧写。

串口发送：状态控制寄存器SCON、COMPIM

**基于c51温度检测报警(模数转换ADC0809、数码管显示、中断)**

模拟：程序比较简单：三个按钮，设置阈值上下限。记录传感器串口连接地址， mcu去地址取数字数据（ADC—>数字）

DS18B20温度计采集，51单片机读取温度并判断，大于阈值调用蜂鸣器的IO口

**基于c51蓝牙循迹小车**

**学习目标**：PWM控速、Uart传输、外设控制、蓝牙协议、安卓蓝牙开发

手机与蓝牙模块连接时（中断），停止小车：手机按下按钮发送某个数字（到中断函数），把蓝牙模块接受数据送到一个接受函数处理，使能相应串口控制舵机

[android蓝牙简单开发\_bluetoothadapter.getdefaultadapter()-CSDN博客](https://blog.csdn.net/qq_41818873/article/details/121745405#:~:text=android%E8%93%9D%E7%89%99%E7%AE%80%E5%8D%95%E5%BC%80%E5%8F%91%201%201%E3%80%81%E6%9D%83%E9%99%90%E7%94%B3%E8%AF%B7%20%3Cuses-permission%20android%3Aname%3D%22android.permission.BLUETOOTH%22%2F%3E%20%20%E4%BD%BF%E7%94%A8%E8%93%9D%E7%89%99%E6%89%80%E9%9C%80%E8%A6%81%E7%9A%84%E6%9D%83%E9%99%90%20...,5%E3%80%81%E8%93%9D%E7%89%99%E9%85%8D%E5%AF%B9%20...%206%206%E3%80%81%E8%8E%B7%E5%8F%96%E5%B7%B2%E7%BB%8F%E9%85%8D%E5%AF%B9%E7%9A%84%E8%AE%BE%E5%A4%87%20...%207%207%E3%80%81%E8%BF%9E%E6%8E%A5%E8%AE%BE%E5%A4%87%20)

**----PWM控制小车运动速度**

1. 准备工作: 循迹小车用到的主要零件有:

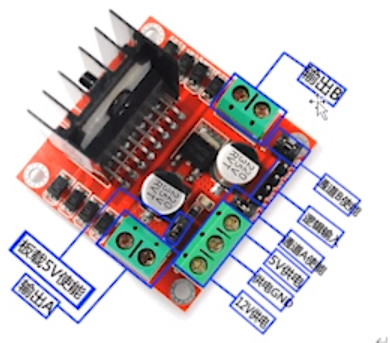
STC89C51单片机、L298N、TCRT5000红外反射传感器、HC-08 蓝牙模块、12V 可充电电池、两个电机和轮子、一个小车底盘、若干杜邦线等。

1. 理论分析:

通过51 单片机的 io 口来控制其他模块，从而来控制小车遇到什么情况对于传回的信号需要做出什么应对措施。在蓝牙控制方面，单片机通过蓝牙和手机进行通信，对手机发出的指令然后做出相应的动作。在寻迹方面，通过红外传感器接感应到轨迹然后转为数字信号传给单片机，使单片机做出相应的动作。“

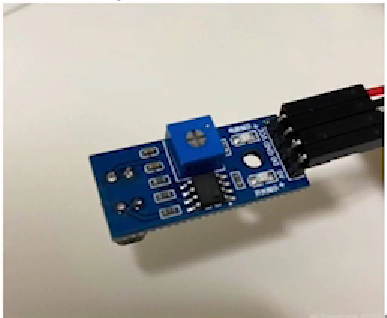
首先分析一下需要用到的模块的功能;

（1） L289N 模块，如下图是我用到的一个 L289N 的实物图



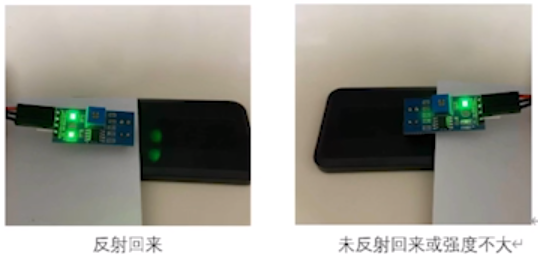
可以用12V的电池来接12V供电口通过该模块来给整个小车供电，其中5V 可以给单片机供电。IN1、IN2、IN3、IN4连接单片机来IO 口，这里我连接的是单片机的P1.0、P1.1、P1.2、P1.3，可以通过单片机来输出高低电平来控制 L289N 模块中四个 OUT 的高低电平，从而来控制两个电机的正转和反转。因为要用到PWM调速（使能口开停），所以这里我取掉了 L298N通道A、B使能的跳线帽，也用单片机来控制（P1.4、P1.5）。“

（2） TCRT5000 红外反射传感器，如下图是我用的实物图

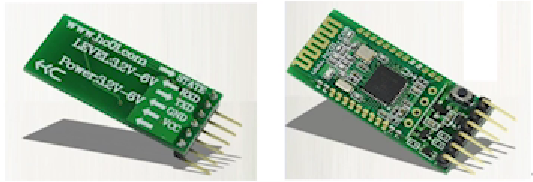


该模块一共有四个引脚，分别为VCC、GND、DO、AO，其中VCC 工作电压为3.3V-5V，但是在实际操作中发现该模块接在单片机的3.3V 的输出口并不能使该传感器正常工作，接在 5V 上可以正常工作。DO:TTL开关信号输出。AO: 模拟信号输出，AO 在这次设计中不用，因此只需要连接DO即可。

TCRT5000 传感器的红外发射二极管不断发射红外线，当发射出的红外线没有被反射回来或被反射回来但强度不够大时，红外接收管一直处于关断状态，此时模块的输出端为高电平，指示二极管一直处于熄灭状态。被检测物体出现在检测范围内时，维外线被反射回来且强度足够大，红外接收管饱和，此时模块的输出端为低电平，指示二极管被点亮。



（3） HC-08 蓝牙模块 (带底板)，实物图如下:



HC-08 (带底板)总共有六个引脚，用到的有 VCC、GND、RXD、TXD这四个引脚，前两个是给 HC-08 供电，后两个RXD 和TXD 与单片机的RXD 和TXD 交叉连接，与单片机行通信。工作电压是 3.2V-6V

在串行通信中，通常会使用两条线路来实现数据的双向传输，即接收线（RXD）和发送线（TXD）。在与单片机（或微控制器）进行串行通信时，通常会将外部设备的 RXD（接收线）连接到单片机的 TXD（发送线），而外部设备的 TXD（发送线）连接到单片机的 RXD（接收线）。这种连接方式称为交叉连接（cross connection）或反向连接（reverse connection），是为了确保通信的正常进行。

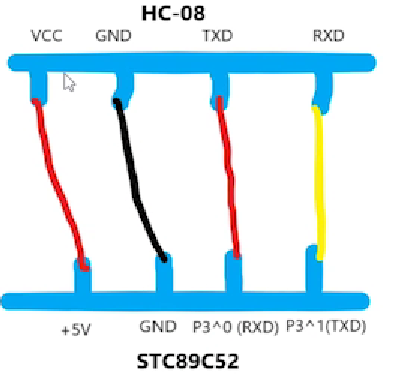
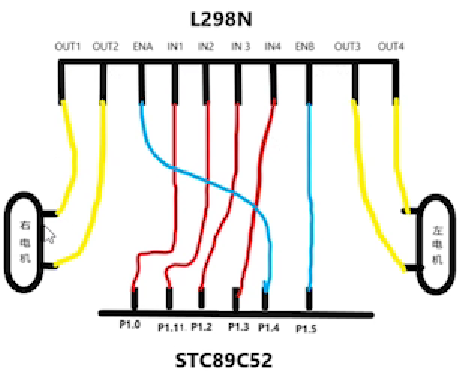
这种交叉连接的原因如下：

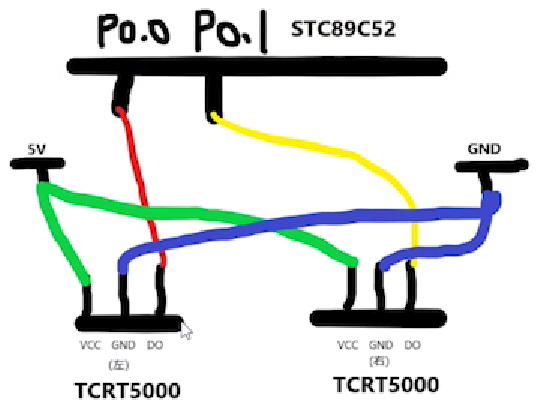
1. \*\*传输方向匹配\*\*：外部设备的发送线（TXD）需要与单片机的接收线（RXD）相连接，以便外部设备发送的数据能够被单片机接收。反之，外部设备的接收线（RXD）需要与单片机的发送线（TXD）相连接，以便单片机发送的数据能够被外部设备接收。

2. \*\*通信协议约定\*\*：在串行通信协议中，通常规定了数据的发送和接收方向，需要严格遵守以确保通信的正确性。例如，通常情况下，UART（通用异步收发器）通信协议规定了发送和接收方向。

3. \*\*单片机硬件设计\*\*：单片机的串行通信硬件通常设计为发送线（TXD）和接收线（RXD），并且单片机通常提供了对应的串行通信引脚，以便连接外部设备。

因此，为了确保串行通信的正常进行，需要将外部设备的 RXD 和 TXD 分别与单片机的 TXD 和 RXD 交叉连接。这样，单片机就能够正确地接收外部设备发送的数据，并将数据发送给外部设备，实现双向通信。





通过这次课堂设计，学会了如何利用单片机与其他模块进行通信，并且学到了很多实践知识。在将理论转化的现实的时候，在蓝牙控制这一功能还算是比较顺利，很快就弄好了。但是在循迹方面花费了很大的功夫，比如起初一直找不到合适的能够使红外传感器识别的赛道，不断的调节红外传感器的识别灵敏度以及更换合适的识别材料（购买的专业吸光胶布)，才慢慢识别，但是又有个问题就是在识别转弯的时候，转弯的时候速度过大或者太慢从而使小车跑出规定的赛道，还是需要不断的调试程序来调速，使小车达到差不多的转弯速度能够正常循迹。看似觉得简单的东西，但是实际做的时候总能够产生很多让人意想不到的状况，所以在以后的学习生涯中要不仅要搞懂理论知识，而且还要注重实践提作，解决自己可能疏忽的地方，使自己的思维更加缜密。“

**定时器作用**

精确计时、延时、脉冲生成、定时中断、计数器

**STM32**

**基于STM32水质检测**

[基于STM32的智能水质监测系统设计-毕业设计-CSDN博客](https://blog.csdn.net/m0_73485055/article/details/132514152)

[STM32单片机通过ESP8266WiFi模块与Android APP实现数据传输（一）---下位机硬件配置\_stm32wifi模块接收手机数据-CSDN博客](https://blog.csdn.net/weixin_44355077/article/details/116121764)

[基于STM32单片机与wifi模块串口结合进行PC端或手机端无线通信（附项目资料包）\_wif配置stm32的参数-CSDN博客](https://blog.csdn.net/LJH_1999/article/details/95028436?ops_request_misc=&request_id=&biz_id=102&utm_term=stm32+wifi+%E6%89%8B%E6%9C%BA&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-95028436.142%5ev99%5epc_search_result_base7&spm=1018.2226.3001.4187)

[【常用模块】ESP8266 WIFI串口通信模块使用详解（实例：附STM32详细代码）\_正点原子wifiesp8266提供代码-CSDN博客](https://blog.csdn.net/qq_38410730/article/details/86538288?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171135048416800184120221%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171135048416800184120221&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-86538288-null-null.142%5ev99%5epc_search_result_base7&utm_term=stm32wifi&spm=1018.2226.3001.4187)

**学习目标**：STM32开发、AT指令、WIFI透传、Usart、I2C、SPI、TCP/IP通信（wifi模块）、串口调试

目前嵌入式使用比较多的无线通信技术有红外遥控、蓝牙、WIFI。红外遥控技术需要端口对接，受障碍物影响较大，而且传输距离较短；而蓝牙技术虽然没有了障碍物的影响，能够全方位地传输信息，但缺点是传输距离仍然较短，传输速率较慢；WIFI 相对以上两者，没有了那些缺点，传输距离和传输速率都有了明显的提升，以上三种技术中显然 WIFI是最适用的

WiFi模块配置

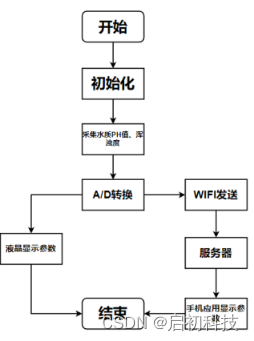
本文使用的WiFi模块具有三种工作模式，分别是：STA模式、AP模式和STA+AP模式，在每个模式下又分了三种模式分别是：UDP、TCP Client 和TCP Server模式，如果不太了解WiFi模块的工作模式，大家可以先去了解一下。

本文将WiFi模块设置成了AP模式下的TCP Server模式，就相当于一个路由器，并配置好WiFi模块的IP地址和端口号，完事就可以通过手机APP连接这个WiFi模块，并相互传输数据，进而就可以控制单片机所连接的硬件设备，实现简易物联网的功能。

WiFi模块配置是通过AT指令完成的。

单片机外接 E-201 系列的 PH 传感器和TSW-30 浑浊度传感器采集数据然后通过单片机计算具体水质数据，再将水质数据显示到 LCD1602 液晶显示器上，并外接 ESP8266WIFI 模块通过单片机将水质数据实时传输到手机应用上显示，该设计与传统的水质检测相比有着采集效率高，实时性好，成本低等优点。同时，水资检测是保护水资源的重要手段之一，对人们的身体健康有着良好的保障，对国家的发展有着重要意义。

本设计以 STM32 为核心，外接液晶显示模块、PH 传感器、浑浊度传感器、WIFI模块实现检测水质 PH 值和浑浊度，STM32 单片机通过 PH 传感器以及浑浊度传感器获取电压信号，然后单片机获取信号再将具体数据计算结果，将结果通过程序显示在 LCD1602，同时单片机利用 WIFI 模块发送水质的 PH 值和浑浊度数据，手机应用获取到具体数据并显示出来，具体程序流程图如图 4.1。开始时需要时钟初始化、IO 口初始化、串口初始化、液晶初始化等。其中获取 PH 值和浑浊度参数是通过单片机的采集电压而来的，然后算出实际数据



**Wifi模块中的透传是什么？**

简单来说，透传模式允许 Wi-Fi 模块在 Wi-Fi 网络和串口之间实现一种透明的数据传输通道，无需对数据进行解析、处理或转换，使得外部设备可以直接与 Wi-Fi 网络进行通信，而无需了解 Wi-Fi 协议的细节。

透传模式通常由 Wi-Fi 模块提供的特定 AT 指令来配置和控制。通过发送特定的 AT 指令，可以将 Wi-Fi 模块配置为透传模式，并设置相关参数，如波特率、串口格式等。一旦配置完成，Wi-Fi 模块就会将串口上接收到的数据直接传输到 Wi-Fi 网络中，同时将 Wi-Fi 网络接收到的数据直接发送到串口上，实现数据的透明传输。

**什么是AT指令？**

AT 指令通常用于串口通信，通过发送特定的 AT 指令到设备，可以控制设备的行为、配置设备的参数、查询设备的状态等。AT 指令在各种通信设备中广泛应用，包括调制解调器、手机、GPS 模块、Wi-Fi 模块等。

**gpio和****usart区别？**

GPIO 是一种通用的数字输入输出引脚，用于与外部设备进行数字信号的交互

USART（通用同步/异步串行接口）\*\*：

- USART 是一种用于串行通信的接口，可以实现同步串行通信（如 SPI）和异步串行通信（如 UART）。

- USART 可以与外部设备进行数据通信，例如与其他设备进行数据传输，或者与计算机进行串口通信。

- USART 通常具有配置波特率、数据位数、校验位和停止位等参数的功能，以适应不同的通信需求。

- USART 通常包含发送（TX）和接收（RX）两个引脚，用于串行数据的发送和接收。

**基于STM32智慧农业终端**

[什么是物联网NB-IoT技术？ - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/344485911)

[大白获取BC26和基站通信的log数据视频教程\_bc26抓取log-CSDN博客](https://blog.csdn.net/u011943791/article/details/82315440)

[基于STM32的智慧农业管理系统设计与实现\_nng stm32-CSDN博客](https://blog.csdn.net/xiaolong1126626497/article/details/136532028)

[NB-IOT (2)---移远BC26模块模组快速入门\_quectel\_bc26-te-b\_用户指导-CSDN博客](https://blog.csdn.net/zhangbijun1230/article/details/82251236)

[MQTT协议，终于有人讲清楚了 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/421109780)

[APP与服务器之间通过 http(POST、GET)进行数据交互 ( 实现一个简单的物联网系统-1 )\_app与服务器http通信-CSDN博客](https://blog.csdn.net/CSDN_X_W/article/details/86750171)

[Android通过http与服务器端交互\_安卓app通过http连接远程服务器-CSDN博客](https://blog.csdn.net/yeyuxp/article/details/11606477?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-11606477-blog-86750171.235%5Ev43%5Epc_blog_bottom_relevance_base5&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-11606477-blog-86750171.235%5Ev43%5Epc_blog_bottom_relevance_base5&utm_relevant_index=2)

[Android操作HTTP实现与服务器通信\_安卓app通过http连接远程服务器-CSDN博客](https://blog.csdn.net/abcd_3344_abcd/article/details/50668497?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-1-50668497-blog-11606477.235%5Ev43%5Epc_blog_bottom_relevance_base5&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-1-50668497-blog-11606477.235%5Ev43%5Epc_blog_bottom_relevance_base5&utm_relevant_index=2)

**学习目标**：STM32开发、NBIoT长距离通信、USB接口log抓取（模块和服务端之间）、GPS（卡中自带）、SSH（pc到服务器）、HTTP（服务器与app）、MQTT（终端到服务器）等协议

Windows下安装 FinalShell 终端，方便使用**SSH协议远程登录到MQTT云服务器**

本系统主要由传感器模块、STM32微控制器模块、无线通信模块和上位机软件组成。传感器模块负责采集农业环境信息，如温度、湿度、光照等；STM32微控制器模块负责数据处理和指令下发；无线通信模块负责将数据传输到上位机软件；上位机软件负责数据的展示和控制指令的下发，并在超出阈值时通过蜂鸣器报警，提醒管理人员进行灌溉、施肥等操作，STM32微控制器解析指令并执行相应的操作。

通过ADC（模数转换器）将传感器输出的模拟信号转换为数字信号，然后进行数据处理和存储。数据处理包括数据滤波、数据校准等步骤，以提高数据的准确性和可靠性

同时，采用NBIoT通信技术（BC26模块）将采集到的数据上传至云端，利用EMQX开源MQTT服务器框架部署于华为云ECS服务器上的MQTT服务器，实现数据的远程展示与处理。

系统支持微信小程序远程控制功能，使得农户或管理者可以随时随地查看农田环境参数、接收预警信息，并能远程手动控制灌溉设备、补光灯等，大大提高了农业生产的智能化和便捷性。此项目的实施不仅有助于推动我国农业信息化水平的提升，也有利于农业资源的高效利用，对保障国家粮食安全、促进农业增效、农民增收具有重要意义。

1. 实时环境监测： （2）自动化管理与预警：

（3）远程控制功能：利用NBIoT通信技术（BC26模块）将现场采集到的各项数据上传至云端MQTT服务器，并通过微信小程序实现远程访问和展示。用户可以通过微信小程序查看实时监测数据，以及对农田设备进行远程手动控制，如启动或关闭5V抽水泵进行灌溉，开启或关闭白色LED补光灯调节光照条件。

（4）数据上云与分析：基于EMQX开源MQTT服务器框架搭建的MQTT服务器，能够接收并处理STM32主控板传输的农业环境数据，并对接微信小程序平台，为用户提供直观易懂的数据图表和分析结果，便于农户或农业技术人员进行科学决策和精准管理。

**8.1 BC26通信模块的AT指令调试过程**

BC20/BC26 开启GPS、连接MQTT服务器的AT指令发送流程。

(1)查询模块是否正常(2)获取卡号,查询卡是否插好(3)激活网络(4)获取网络激活状态(5)查询网络质量(6)检查网络状态(7)激活GPS(8)查询GPS激活状态(9)获取一次GPS定位语句(10)连接MQTT服务器(11)登录MQTT服务器(12)订阅主题(13)发布主题

**USB接口转双串口芯片**、**主串口和调试串口接口处有LED信号指示灯**。用**microUSB可以获取到BC26和基站通信的log数据**。BC26所有1.8V的IO口都转换成了3.3V的电平,开发板留出了BC26的3个调试串口、所有GPIO口、I2C接口、SPI接口、2个GPIO控制的LED灯、电源指示灯、休眠唤醒接口、BC26电源启动接口、BC26复位接口、3.3V和5V电压输出接口、GND接口。

**ZigBee**

**基于stm32和ZigBee智能家居（温湿度+灯光开关）**

**学习目标**：ZigBee组网、短距离通信、网络编程、wifi模块、JTAG调试。

[基于ZigBee的智能家居系统设计-CSDN博客](https://blog.csdn.net/qq_44717552/article/details/125712621?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171136999916800192273204%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171136999916800192273204&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-1-125712621-null-null.142%5ev99%5epc_search_result_base7&utm_term=%E5%9F%BA%E4%BA%8Ezigbee%E7%9A%84%E6%99%BA%E8%83%BD%E5%AE%B6%E5%B1%85&spm=1018.2226.3001.4187)

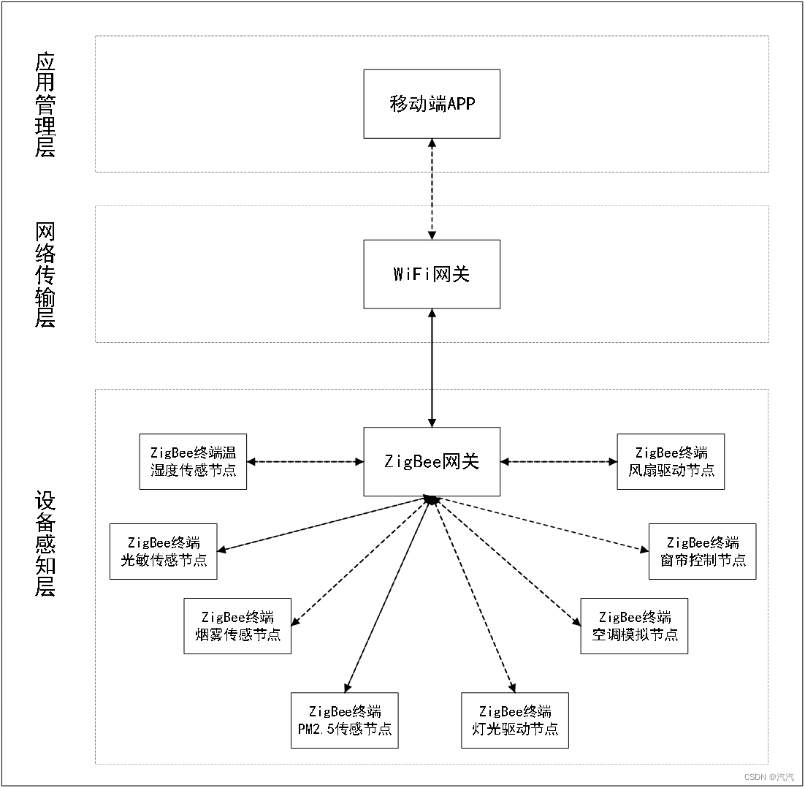
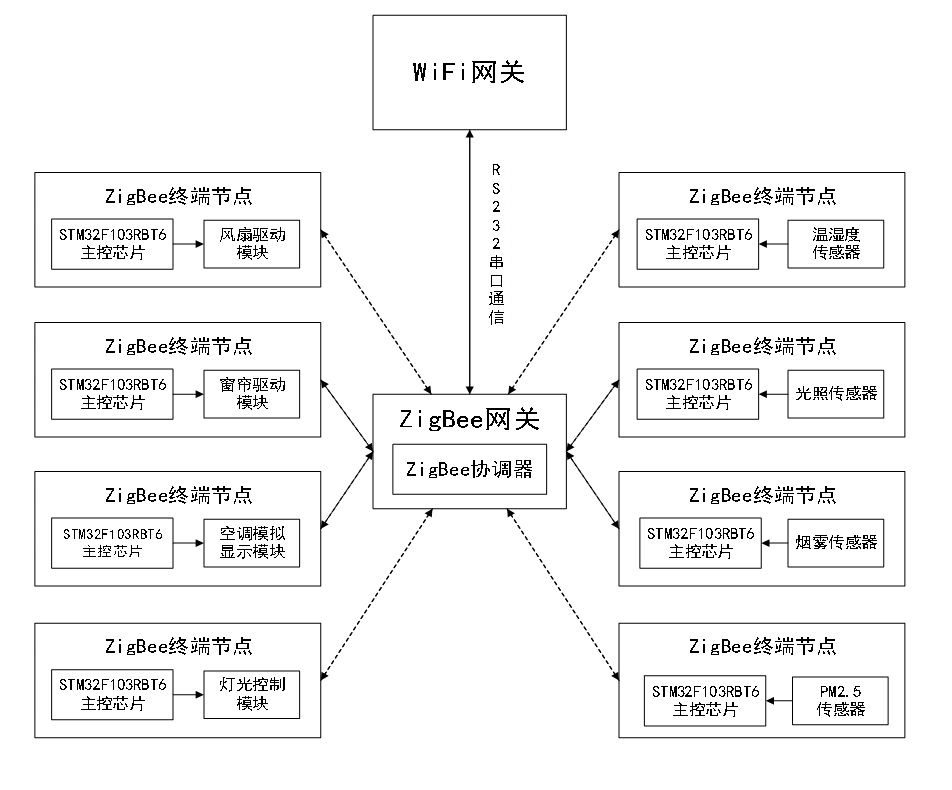
[Zigbee模块（CC2530）详解-CSDN博客](https://blog.csdn.net/m0_56694518/article/details/131483542#1.%20%E5%B8%B8%E8%A7%81%E7%9A%84Zigbee%E6%A8%A1%E5%9D%97)

[J-Link仿真器与JTAG和SWD下载与接线介绍\_swd和jtaj下载仿真-CSDN博客](https://blog.csdn.net/as480133937/article/details/96627050)

[11\_zigbee组网过程分析\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV14K4y1k7dQ/?spm_id_from=333.337.search-card.all.click&vd_source=c4b26319211bc9ccc5ba3a0a22916e39)

socket网络、局域网、wifi、USB

1. 先用一个STM32接入WiFi网关节点模块设为服务端用于接收手机发送的数据（TCP/socket）：创建socket、监听、接受监听。
2. 服务端在接一个ZigBee模块作为（协调器）ZigBee网关
3. 再用一个ZigBee作为终端，与协调器和Stm32主控芯片连接。Stm32外接温湿度传感器步进电机模块

USB 下载和 JTAG 下载都是常见的单片机固件下载方法。USB 下载适用于快速、简单地将固件下载到单片机中，而 JTAG 下载适用于需要调试和测试功能的开发和调试阶段。JTAG 下载通常需要使用支持 JTAG 接口的下载器或调试器，如 JTAG ICE、J-Link 等

**组网过程：**

在此之前，了解了 zigbee 的通信过程，以及组网 **1.zigbee网络中有4种通信方式：**

单播通信，与指定设备通信

组播通信，与组内设备通信

广播通信，与网内设备通信

绑定通信，不是设备之间通信，而是不同端点之间的通信

**2.zigbee网络中有3种通信设备：**

全功能设备FFD：协调器

全功能设备FFD：路由器

半功能设备RFD：终端

接下来我们具体了解一下 zigbee 是怎样组网的，组完网络后的通信过程

**1.第一步先通过底层能量扫描，选择一个合适的信道建立**

**2.第二步协调器或路由器会不断广播信标（Beacon）**

**3.第三步设备或路由器发现信标，请求入网**

**4.协调器或路由器会分配短地址给设备或路由器** // 观看路由器是要假如网络还是加入别的节点，协调器和路由器也有自己的短地址，短地址是一个会变化的地址入网成功

**5.基于端点展开通信**  //每次通信不论发送还是接收，都要创建一个端点，就可以展开通信

**6.协调器如何完成端点的扫描，和端点的建立**

**STM32使用CC2530模块**

**硬件连接：**

连接CC2530模块的VCC引脚到STM32的适当电源引脚，并确保它们具有共同的地引脚（GND）连接。

连接CC2530模块的TX引脚到STM32的UART或SPI总线的RX引脚，将CC2530模块的RX引脚连接到STM32的UART或SPI总线的TX引脚。如果使用SPI接口，则还需要连接CC2530模块的CLK和CS引脚到STM32的对应引脚。

**软件设置：**

在STM32的开发环境中创建一个新的工程，并为所选的通信接口（UART或SPI）配置相应的引脚。在STM32的开发环境中，配置UART或SPI外设的时钟和参数，例如波特率、数据位数和停止位数等。确保它们与CC2530模块的配置相匹配。

**通信协议：**

根据所选的通信接口（UART或SPI），编写相应的驱动程序或库函数，以实现STM32与CC2530模块之间的数据交换。

如果使用UART通信，您需要实现相应的UART接收和发送函数，以便与CC2530模块进行数据交换。

如果使用SPI通信，您需要实现相应的SPI接口函数，包括发送和接收数据的功能。

**uart和usart的区别**

[UART与USART区别 (串口同步通信和异步通信)\_uart同步和异步的区别-CSDN博客](https://blog.csdn.net/lyd0813/article/details/109499901)

[STM32+Zigbee模块实现串口通信获取传感器数据\_zigbee模块通信-CSDN博客](https://blog.csdn.net/weixin_57246604/article/details/126918811)

UART与USART都是单片机上的串口通信，他们之间的区别如下：

首先从名字上看：

UART：universal asynchronous receiver and transmitter通用异步收/发器

USART:universal synchronous asynchronous receiver and transmitter通用同步/异步收/发器

从名字上可以看出，USART在UART基础上增加了同步功能，即USART是UART的增强型，事实也确实是这样。但是具体增强到了什么地方呢？

其实当我们使用USART在异步通信的时候，它与UART没有什么区别，但是用在同步通信的时候，区别就很明显了：大家都知道同步通信需要时钟来触发数据传输，也就是说USART相对UART的区别之一就是能提供主动时钟。

1. \*\*UART（通用异步收发器）\*\*：

- UART 主要用于实现异步串行通信，不依赖外部时钟信号，而是通过发送端和接收端预先约定的波特率来进行数据的传输。

- UART 通常包含两个引脚：发送（TX）和接收（RX），分别用于发送数据和接收数据。

2. \*\*USART（通用同步/异步串行接口）\*\*：

- USART 是通用同步/异步串行接口的简称，它是一种更加通用的串行通信接口，可以实现同步串行通信（如 SPI、i2c）和异步串行通信（如 UART）。

- USART 可以根据需要配置为同步模式或异步模式，因此具有更大的灵活性和功能性。

- USART 通常包含发送（TX）、接收（RX）、时钟（CLK）和其他控制信号，可以支持更多的通信协议和功能。

二. 软件设计也存在同步发送和异步发送的区别:

同步是指：发送方发出数据后，等接收方发回响应以后才发下一个数据包的通讯方式。

异步是指：发送方发出数据后，不等接收方发回响应，接着发送下个数据包的通讯方式。

同步是阻塞模式，异步是非阻塞模式。

三. 单工\半双工\全双工

单工、半双工、全双工 单工数据传输只支持数据在一个方向上传输； 半双工数据传输允许数据在两个方向上传输，但是，在某一时刻，只允许数据在一个方向上传输，它实际上是一种切换方向的单工通信； 全双工数据通信允许数据同时在两个方向上传输，因此，全双工通信是两个单工通信方式的结合，它要求发送设备和接收设备都有独立的接收和发送能力。

I2C是半双工，SPI的全双工，uart是全双工。

SPI：SPI 全称为 Serial Peripheral Interface（串行外围设备接口）。

I2C：I2C 全称为 Inter-Integrated Circuit（双向串行总线）。

长城汽车

车载软件开发-音频方向

了解ARM架构、了解高通音频架构、了解QNX系统开发、了解CAN和FDbus总线。熟悉车载报警系统avas开发，熟悉基于CAN对ECU节点的调试。

/Rots/ QNX

[组成车载OS的三大阵营：QNX、Linux、Android - 21ic电子网](https://www.21ic.com/embed/software/linuxos/201912/92472.html)

[DSP基础知识回顾-CSDN博客](https://blog.csdn.net/m0_56561130/article/details/123813239?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171142375216800197050766%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171142375216800197050766&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-123813239-null-null.142%5ev99%5epc_search_result_base7&utm_term=DSP&spm=1018.2226.3001.4187)

[嵌入式中的BSP---BSP到底是什么？\_嵌入式bsp是什么意思-CSDN博客](https://blog.csdn.net/shangtang1/article/details/79248947?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171142897816800197022447%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=171142897816800197022447&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-4-79248947-null-null.142%5ev99%5epc_search_result_base7&utm_term=dsp%E5%92%8Cbsp%E5%88%86%E5%88%AB%E6%98%AF%E4%BB%80%E4%B9%88&spm=1018.2226.3001.4187)

[嵌入式系统BSP基础知识\_嵌入式bsp-CSDN博客](https://blog.csdn.net/usstmiracle/article/details/121767788?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPayColumn-1-121767788-blog-79248947.235%5Ev43%5Epc_blog_bottom_relevance_base5&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPayColumn-1-121767788-blog-79248947.235%5Ev43%5Epc_blog_bottom_relevance_base5&utm_relevant_index=1)

[高通SA8155原理图和特性介绍（1）\_骁龙8155参数-CSDN博客](https://blog.csdn.net/weixin_49775512/article/details/132398512?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-132398512-blog-130087975.235%5Ev43%5Epc_blog_bottom_relevance_base5&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-132398512-blog-130087975.235%5Ev43%5Epc_blog_bottom_relevance_base5&utm_relevant_index=5)

[Android automotive车载开发（1）-----Automotive audio\_安卓aaos-CSDN博客](https://blog.csdn.net/LJX646566715/article/details/127547812?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171142939116777224451362%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171142939116777224451362&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-127547812-null-null.142%5ev99%5epc_search_result_base7&utm_term=%E8%BD%A6%E8%BD%BD%E9%9F%B3%E9%A2%91%E5%BC%80%E5%8F%91&spm=1018.2226.3001.4187)

[基于MCU+DSP车载音频系统开发调试流程\_汽车功放mcu和dsp-CSDN博客](https://blog.csdn.net/xinbo7077/article/details/125416144?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_utm_term~default-4-125416144-blog-127547812.235%5ev43%5epc_blog_bottom_relevance_base5&spm=1001.2101.3001.4242.3&utm_relevant_index=7)

[高通音频架构（三）\_adsp-CSDN博客](https://blog.csdn.net/wh2526422/article/details/126247382)

<https://www.zhihu.com/question/35630289>

<https://d1.amobbs.com/bbs_upload782111/files_29/ourdev_553420.pdf>

[Fast Distributed Bus - FDBus：高速分布式总线以及中间件开发框架\_fdbus:高速分布式总线-CSDN博客](https://blog.csdn.net/jeremy_cz/article/details/89060291)

音频处理Audio和音频处理接口：Adsp可以处理基于A2B的音频信号，音频流可以通过TDM传输入SOC。

南京拓界

安卓、USB