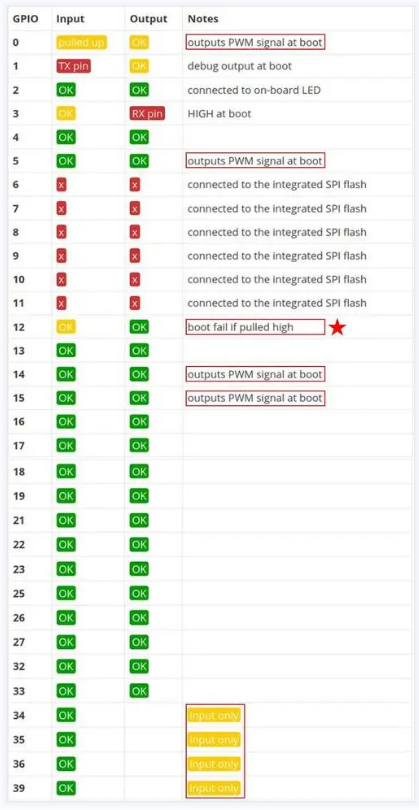
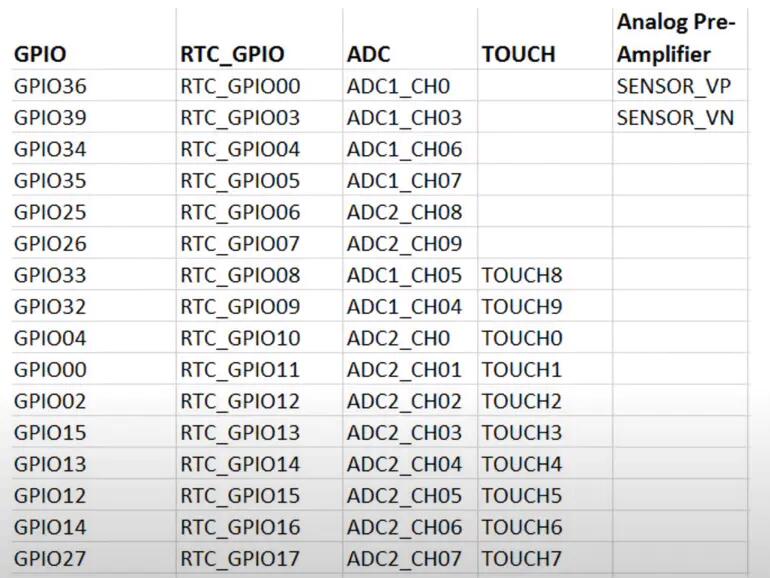
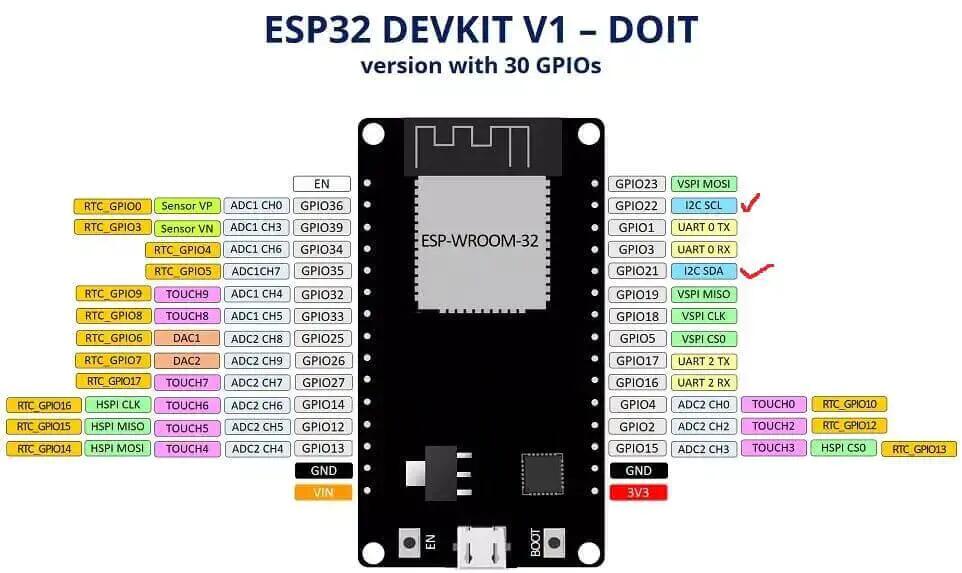
Learn ESP32 with arduino 笔记

官方WIFI库：https://www.arduino.cc/reference/en/libraries/wifi/

1. **Getting to know your ESP32**
2. 含一个双核CPU，可达8160或240兆赫
3. 含一个超低功耗的协处理器
4. 含有大量内存，甚至支持电路板上的外部内存
5. 内置蓝牙与WIFI
6. 含一个12位ADC（模拟输入）来测量外部电压
7. 含10个触摸传感器检测电容
8. 含一个LED电源管理芯片
9. 含一个内置的霍尔效应传感器加密加速
10. 含34个可编程的端口
11. 支持arduino框架，适用于所有的arduino库
12. 有深度睡眠功能，大大延长寿命，被激活时消耗电流小
13. **The new ESP32S2 beta first look**
14. ESP32S2相比于ESP32性能更好一些，但大小变化不大
15. **ESP32 is the best arduino replacement**
16. 不自带闪存，需要电路板上安装
17. 可以在电路板上安装一个USB接口用于给电池充电，通电之后才能使用
18. 电路板上要带有电压调节
19. 可以远程连接互联网
20. 可以连接wifi向服务器发送请求
21. 本身也可以充当服务器
22. 允许两个ESP32之间的连接
23. **Call the best IOT board with multiple connectivity options ESP32 SIM800L**
24. 内置了SIM800L和DP artists，意味着不仅仅是蓝牙和WIFI，还支持短信，电话，GPRS，互联网
25. 作者通过使用API从网站上审查印度的时间和日期做了应该实时和时期的项目
26. 可以摆脱路由器的限制制作一个可以上网的天气站
27. 可以通过物联网打电话发送短信、
28. Arduino程式的安装可以参考以下网址

https://www.bilibili.com/video/BV164411J7GE?spm\_id\_from=333.337.search-card.all.click

**5.Esp32 PINOUT V1 DOIT**



1. 绿色标注的可以使用
2. 黄色标注的可以作为工具
3. 红色标注的不建议作为输入或输出端口
4. PIN0可以使用，可以输入输出，输出的是PWM信号
5. PIN2可以输入输出，通常与LED相连用于测试代码或者基本函数
6. PIN3可以输入但不能输出
7. PIN4,PIN5可以用作输入和输出，PIN5输出PWM信号
8. PIN6-PIN11不能用作输入和输出
9. PIN12可以用作输入和输出，但用作输入时要注意温度
10. PIN13-PIN33可以用作输入和输出，PIN14-PIN15输出PWM信号
11. PIN34-PIN39只允许输入
12. 一个引脚不止一个用法，具体可以参考以上的三张参考图

**6.ESP32 vs Arduino servo motor control**

1. ESP32具有更多的GPIO引脚

2. ESP32有一个servo库，有一些对servo和motor示例代码

3. 将ESP32加载到arduino中时就内置了库

4. ESP32允许对PWM信号使用更高的分辨率

**7. Arduino coding（arduino程式安装）**

具体可以参考给出的链接：

https://www.bilibili.com/video/BV164411J7GE?spm\_id\_from=333.337.search-card.all.click

**8.ESP coding PART1 define variables**

1. 引入一些库：<WiFi.h><PubSubClient.h>

深入学习Arduino PubSubCilent MQTT库：CSDN

https://blog.csdn.net/dpjcn1990/article/details/92831686

2.给出一些必要的数据：SSID（Service Set Identifier：路由器发送的无线信号的名字），password（密码），mqtt\_server（MQTT服务器）

3.定义一些变量：MQTT端口，用户名称，密码，串行通道

4.安装WiFi功能：在工具管理库内搜索，然后下载并安装。运行之后是可辨认的

**9.ESP coding PART2 WIFI and MQTT**

1. 设置WIFI连接：一定的延迟；一些链接成功的提示；SSID

2. 设置一个while循环（!=WIFI\_CONNECTED），反复尝试连接知道成功（需要一个delay延迟）

3. 可以调用WIFI函数（WiFi.localIP()）来获取IP地址

4. 设置一个重新连接的函数

5. 设置一个字符串用来储存已经连接上的用户ID，避免重复

WIFI连接：https://blog.csdn.net/qq\_43482790/article/details/114833239

1. **ESP coding PART3 read incoming data from Arduino**
2. 为串行通信设置一个超时时间，半秒左右
3. 调用WIFI函数
4. 调用客户端服务器参数
5. 调用重新连接函数
6. 调用publish函数发布数据
7. 接受的数据要先放在bufferData（缓冲区数据）内，size可定为500

Bufferdata：<https://blog.csdn.net/wjlgryx/article/details/83974282>

1. 调用memset函数清空缓冲区的数组

Memset()：<https://blog.csdn.net/hk121/article/details/81095797>

1. 调用readBytesUntil函数读取缓冲区的数据

readBytesUntil():

http://www.taichi-maker.com/homepage/reference-index/arduino-code-reference/stream/readbytesuntil/

1. 最后发布数据至频道

**11.BLE server and client communication**

BLE（Bluetooh Low Energy）蓝牙低能耗技术

1. 在两个ESP32之间进行蓝牙信号连接，需要一个作为服务端，一个客户端
2. 导入库BLEDevie.h BLEServer.h BLEUtiles.h BLE2902.h

# 使用Arduino开发ESP32（22）：蓝牙作为客户端使用:https://blog.csdn.net/Naisu\_kun/article/details/116145310

# Arduino ESP32 BLE蓝牙和安卓端蓝牙数据交互实验:<http://www.baidu.com/link?url=zoO-ikGevPZrhkbglxIhbSB59nr01RuuFXsfwhdTRWcukNFf8rWlU_iGFW4nIH8y17x4s0yHiV8ROXtx_rtrVspY0p4y6-hevxDO86JBQra>

# 使用Arduino开发ESP32（十六）：ESP32 BLE Server:https://blog.csdn.net/Aspiration\_H/article/details/107706523

1. 分配PulseSensorpurplepin一个引脚
2. 设置一个LED引脚
3. 创建一个变量信号存储读数并设定一个阈值
4. 定义SERVER\_UUID和Characteristic\_UUID
5. 定义MyServerCallbacks为连接和未连接时的deviceConnected定义布尔值

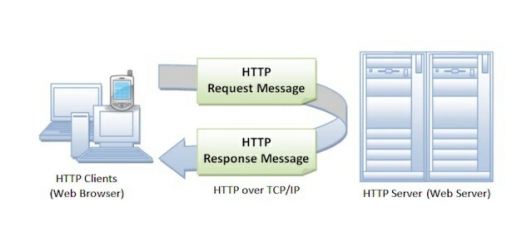


9.



微信小程序蓝牙与ESP32进行配网及数据设置:https://www.cnblogs.com/GeGeBoom/p/13187387.html

1. **Establishing wifi connected with the thing**
2. 导入库，分配SSID和password
3. Setup代码：设置串行波特率为115200
4. 连接至WIFI
5. WIFI.begin功能：用SSID和password连接到网络
6. 定义一个循环，若未连接则持续打印“.”直到连接成功；若连接到WIFI则打印已连接和IP地址
7. 运用串行监视器检测是否成功连接
8. WIFI控制原理



1. 设置服务器传入端口WIFIServer
2. 运用server.begin()告诉服务器开始侦听传入的连接
3. 上传代码并打开串行监视器，确保东西和你电脑连接到一个WIFI
4. 打开浏览器复制这个东西的IP地址到地址栏——向服务器发出一个HTTP请求，如果服务器响应则在loop()内可以进入if(client)
5. 设置一个空字符串来抓住进来的来自客户端的数据
6. 如果有字节用于客户端阅读，就阅读字节并且存储；如果得到换行符，则说明从浏览器收到了一个HTTP请求
7. 进行检测，如果字符串长度为0，则说明收到的HTTP请求已经结束
8. 可以发送一个带有HTML形式的数据的HTTP响应给客户端，这样就可以查看它在浏览器HTTP头宗师以特定的响应代码开始，包括了正在使用的HTTP版本和状态代码
9. 留出一条空行表示标题的结束
10. 输入具体代码
11. **Understanding RSSI and Measuring signal strength**
12. RSSI(Received Signal Strength Indicator)是接收信号的强度指示，用来测量已连接设备接收信号的强度
13. 1mW=0.001W=0dBm
14. 大于1mW的信号有正dBm值，小于1mW的信号有负dBm值
15. WIFI连接信号通常低于1mW，故dBm值通常为0~-100
16. 在WIFI库内有一个WiFi.RSSI()函数，可以获取连接到路由器的RSSI值，返回的值是一个存储在这个变量中的整数值
17. 通常RSSI值低于-90是被认为不可用的
18. **Create MQTT server account**

MQTT中文网：<http://mqtt.p2hp.com/>

# B站 迈向物联网第一步——MQTT理论知识详解:

# https://www.bilibili.com/video/BV1oJ411176Q?spm\_id\_from=333.337.search-card.all.click

# B站 mqtt协议+esp8266+物联网开发:

<https://www.bilibili.com/video/BV1XJ411A7Co?spm_id_from=333.337.search-card.all.click>

1. **Upload code to Arduino and test it**