上手学习路由器漏洞–使用腾达F6

第一篇: 关于TTL

特别感谢:非常感谢KHG小组 怪才 大哥对我关于硬件方面的指点，没有他就没有这篇总结介绍型的文章。

说明：这篇文章介绍的依然很基础，因为我也是个刚弄硬件不久的小白，读者有可能在网上或者书上见到类似的文字，但是那毕竟都是老手高手所做，小白不亲自试一把咋能知道到底怎么搞，我们以前学软件逆向不也是一个道理吗

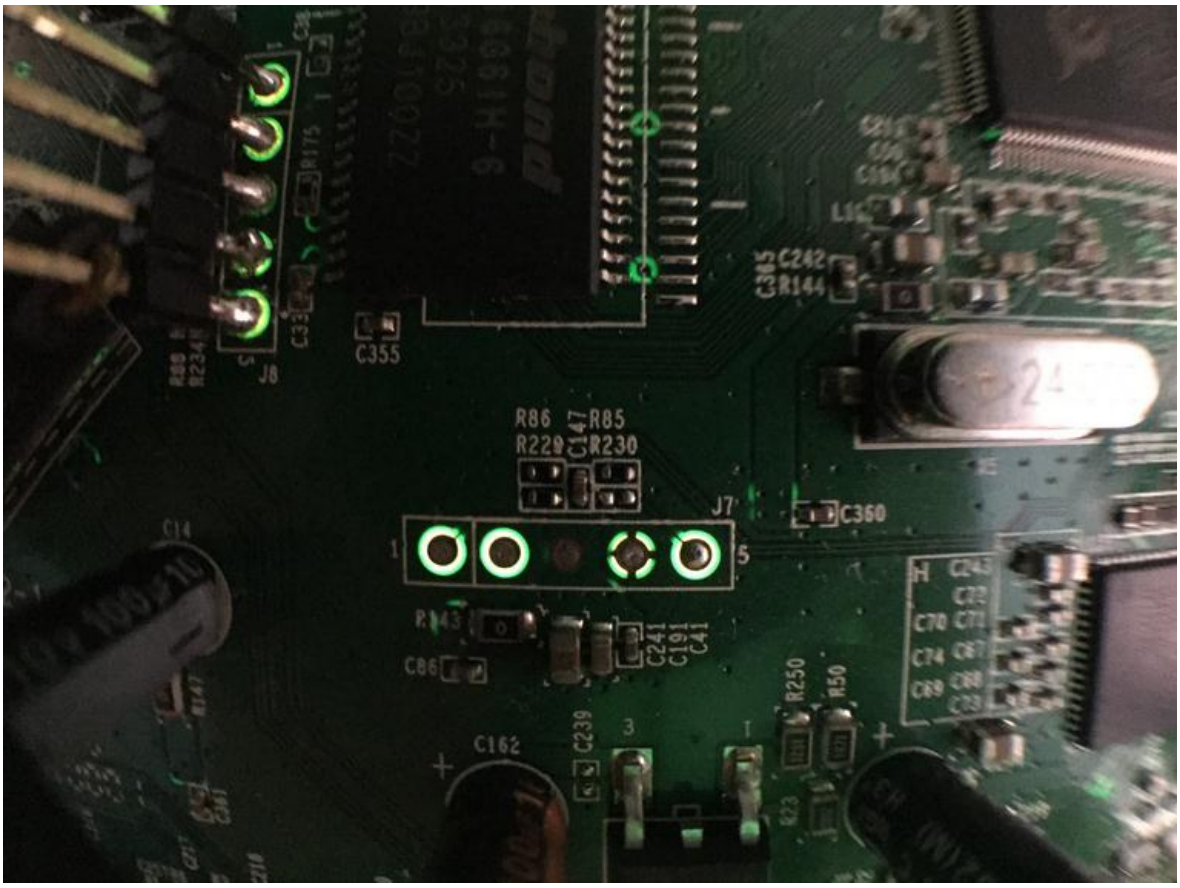
0x0 外观吐槽：(纯吐槽，不想听笔者啰嗦的直接跳到 0x1)

这么诡异的外观，我只能说丑的黔驴技穷，不过挺好拆，连个螺丝都没有

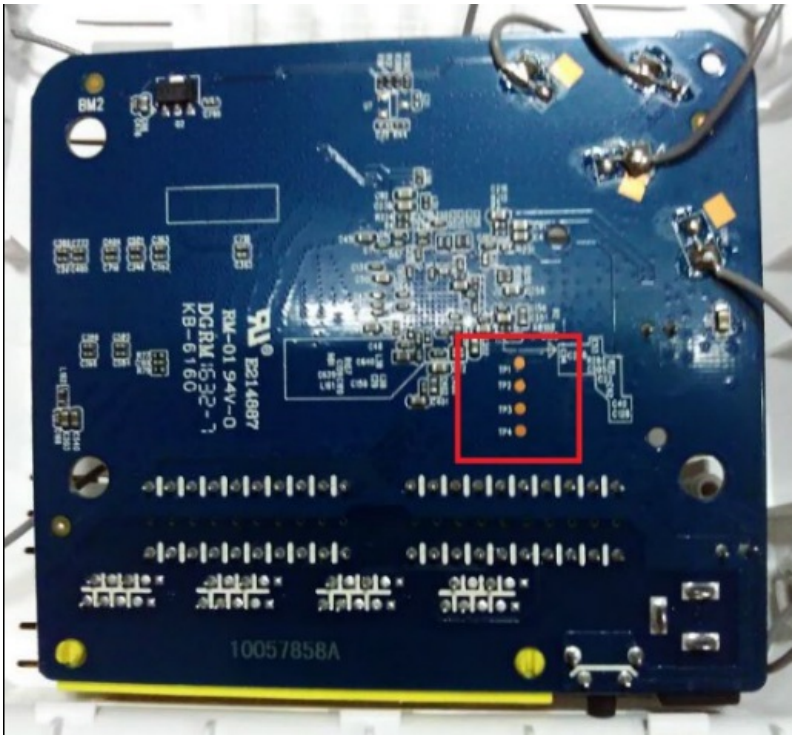
0x1 寻找串口

直接拆开看电路板，找那传说中的串口

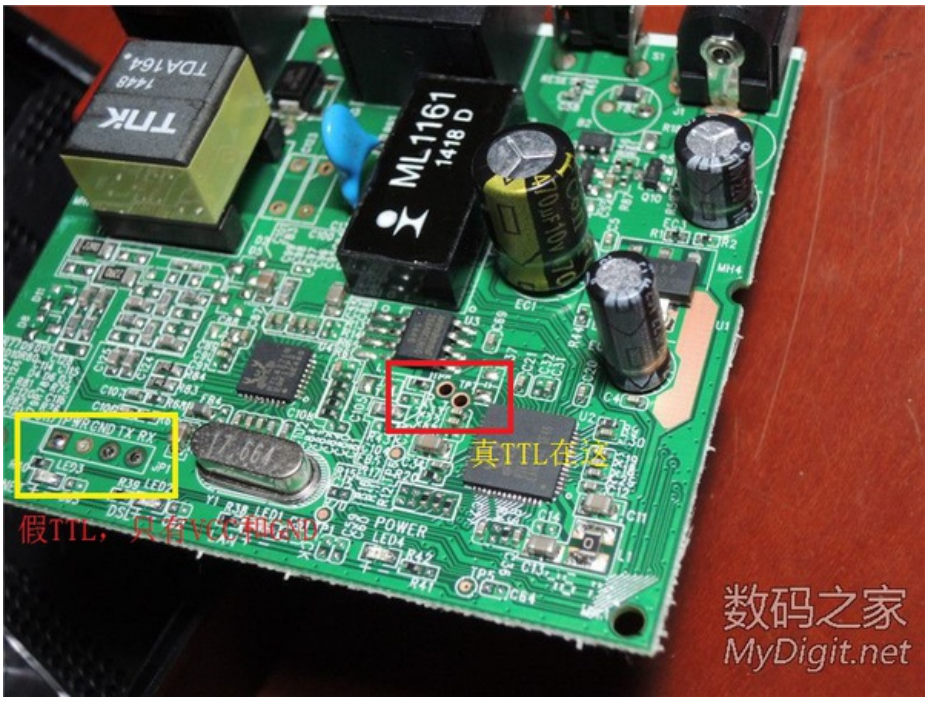
我们在这呢并不能找到如下图所为的五个一排的串口接口



我也不知道为啥，我手里的几个路由都没有这种很明显很好整的串口
那就反过来看看



反面看到我用红色框框圈起来的四个点，他们的标记分别是TP1,TP2,TP3,TP4,经过怪才大哥的指点，这里的TP的意思是test point(跟着大哥涨姿势),应该就是我们找的串口了
这里再说一下，有的呢你看见所谓的四五个一排焊点，甚至有一个焊点是方形的，也不见得就是真实的串口，我在知乎上看到了一个回答说到了这一点，上个图感受下



不过我觉得和我一样的小白碰到这种情况也不比慌，多试几次可疑的点，应该就能出来，当然硬件高手一眼就能看出来，所以我们还要努力学习硬件设计方面的知识，有的同学会说会不会真的没有预留串口接口的呢？有，但是好像不多哦，再套一句怪才大哥的说法：不留测试点不是给自己找麻烦吗？

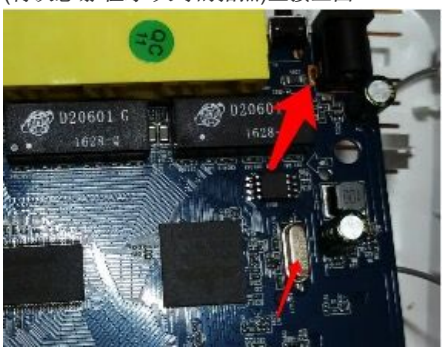
0x2 探测串口引脚

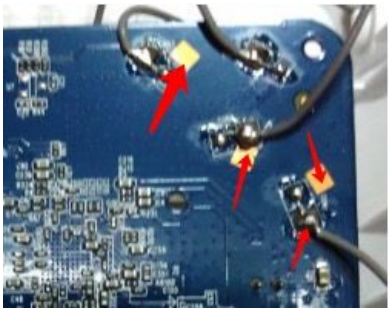
一般UART用下面这个 vcc 电源 GND 接地 TXD 发送信息 RXD 接受信息 我用的万用表，其实用逻辑分析仪更方便，因为穷学生买不起逻辑分析仪，只能用万用表来了。 这里我们不需要连接VCC，但是呢，VCC测出来就可以排除一个，这个也蛮好测，

这里有三点要说的:

1. 电路板有所谓OSP工艺，其实就是在裸铜上面度了一层保护膜，如果我们要上万用表，必须得把这玩意儿刮下去

2. 我们用万用表测试UART的GND，需要有找到在电路板上找到已知的GND，一般大面积覆铜区域和电源插头处会有接地，当然晶振外表也可能接地(再次感谢 怪才 大哥的指点)直接上图





红色箭头都是已知的GND

3. 这种TestPoint太脆弱，很容易搞坏，所以不打孔了，直接焊杜邦线

测试GND

直接用万用表最小电阻档位，挨个试这几个测试点，发现只有tp4显示电阻为0，确定是GND

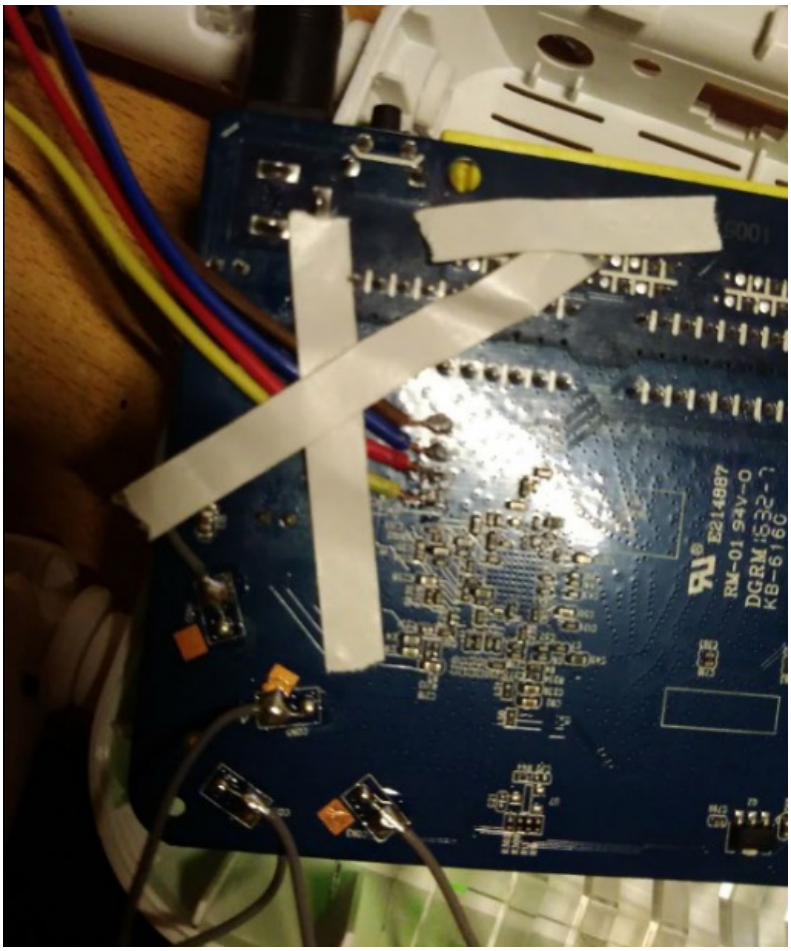
测试VCC，用最小电压档位，给路由器上电，发现tp1加点以后基本就是3.38V左右，确定tp1是VCC

测试TXD 依然是用最小电压，给路由器上点，这里在启动过程中，tp3电压是变化的，看图

剩下的tp2就是rxd了。
其实就算是测不出rxd和txd，直接拿usb转ttl连上gnd,txd.rxd，多试几次就行了，不过因为不知道波特率，可能排列组合次数就多了。

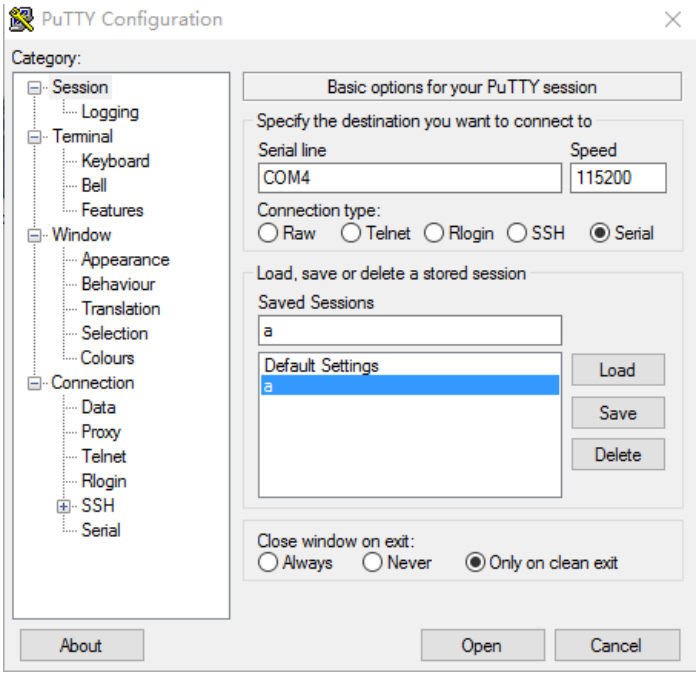
0x3 连接电脑

焊接过程就不说了，焊完了上个图吧，我焊点自己都觉得难看，不过焊的上不连焊能用就行了,我连着VCC也焊上了，就为了练一练 :)，我用双面胶固定线，业余。。。。

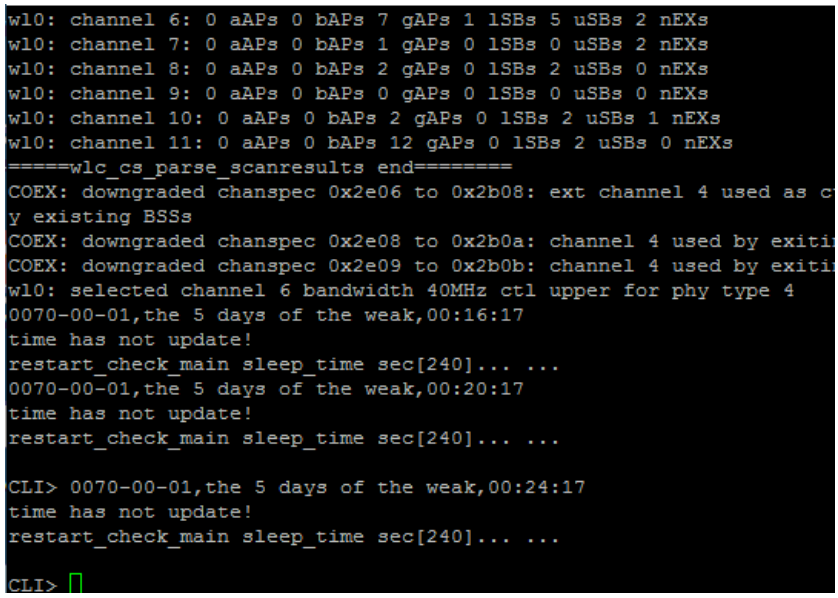


连接次序,网上都有咋连的，这里也写一下吧
GND<-->GND
TXD<-->RXD
RXD<-->TXD
我用这个usb转ttl小板,按照上面的规则链接好了，插到电脑上

关于波特率，可以用《路由器0day》这本书里提供的一个py脚本，但是我懒得开虚拟机了，就用putty，一个一个试，第一个115200就成功了，，貌似不少路由都是这个波特率



直接打开，给路由器上电，就能看到串口的数据了


wl0: channel 6: 0 aAPs 0 bAPs 7 gAPs 1 lSBs 5 uSBs 2 nEXs
wl0: channel 7: 0 aAPs 0 bAPs 1 gAPs 0 lSBs 0 uSBs 2 nEXs
wl0: channel 8: 0 aAPs 0 bAPs 2 gAPs 0 lSBs 2 uSBs 0 nEXs
wl0: channel 9: 0 aAPs 0 bAPs 0 gAPs 0 lSBs 0 uSBs 0 nEXs
wl0: channel 10: 0 aAPs 0 bAPs 2 gAPs 0 lSBs 2 uSBs 1 nEXs
wl0: channel 11: 0 aAPs 0 bAPs 12 gAPs 0 lSBs 2 uSBs 0 nEXs
=====wlc_cs_parse_scanresults end========
COEX: downgraded chanspec 0x2e06 to 0x2b08: ext channel 4 used as c
y existing BSSs
COEX: downgraded chanspec 0x2e08 to 0x2b0a: channel 4 used by exiti
COEX: downgraded chanspec 0x2e09 to 0x2b0b: channel 4 used by exiti
wl0: selected channel 6 bandwidth 40MHz ctl upper for phy type 4
0070-00-01,the 5 days of the weak,00:16:17
time has not update!
restart_check_main sleep_time sec[240]... ...
0070-00-01,the 5 days of the weak,00:20:17
time has not update!
restart_check_main sleep_time sec[240]... ...
CLI> 0070-00-01,the 5 days of the weak,00:24:17
time has not update!
restart_check_main sleep_time sec[240]... ...
CLI>