# 2018/10/26

1.今天也是4点多就起床了，为自己点赞。

2.把尚硅谷的git视频看完了，多谢视频老师：creathinFeng。原来除了github，还有gitlab是专门做私有的，不开源的，估计和码云差不多。肚子饿了，弄点东西吃先。

3.找到个linux视频学一下，貌似评价还不错。尚硅谷-韩顺平

4.点赞获取到了尚硅谷的一套系列教程，不过貌似不怎么适合我。

5.自己写编码转换键值。

6.gbk转键值，发现中文输出有问题，ascii字符没问题。待查找原因

# 2018/10/27

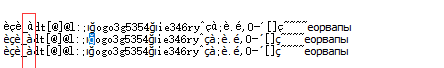
1. 今天睡了懒觉，6点多才醒，然后就不想起来了。估计是昨晚晚睡了（11点多），没听到闹钟响，或者迷迷糊糊就关掉了。罪过
2. 昨天可能钻牛角尖了，中文显示出问题，老以为是接收端的问题。经过一步步排查，觉得接收端没问题，回到发送端看看发送数据，才知道出错了。因为本来发送端有个utf8转gbk的处理，这里讲gbk编码转换出错。已注释掉。
3. 长码显示还有点问题，需要优化一下。
4. 突然发现饶了一大圈有回到了原点。原本的设计就是发送码值，然后接收端根据码值转换成键值。不过需要在发送端把utf82gbk的注释掉。

# 2018/10/28

今天0点后，在琳琳爸妈房间，把礼金也给了。现金9999，支付宝转账给叔叔40000.感觉身体被掏空。接下来就是慢慢还钱了。其实也只是数字而已。

# 2018/10/29

1. 今天5点多起床，也看了几集视频，又是新的一天。
2. 开始测试unicode编码的输出问题，发现个比较难解释的问题。截图如下：



上面有两个形式的a。上面原因也说不清楚。先记下来。

1. 需要找出各个国家的特殊字符，进行测试：

⇠⇢⇡⇣↞↠↟↡←→↑↓↔↕⇄⇅↢↣⇞⇟↫↬⇜⇝↚↛↮↭⇦⇨⇧⇩▲►▼◄➔➙➨➲➜➞➟➠➤➥➦➧➵➸➼➽➺➳↷↶↻↺↵↯➾

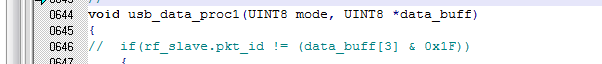
⇠⇢⇡⇣↞↠↟↡←→↑↓↔↕⇄⇅↢↣⇞⇟↫↬⇜⇝↚↛↮↭⇦⇨⇧⇩▲►▼◄➔➙➨➲➜➞➟➠➤➥➦➧➵➸➼➽➺➳↷↶↻↺↵↯➾

以上是箭头类的测试，第一行手写出来的，第二行是扫码显示出来的，没问题。

1. 根据已有的程序，制定较合适的流程图，目前正在修改源程序，明天应该可以测试。

# 2018/10/30

1. 开始调试收发双方，目前还有问题，正在逐步查找。
2. 将定时器的 //rf\_slave.pkt\_id = INIT\_ID; 该行注释，可以输出，但是偶尔成功，也没上传。
3. 上传函数将if语句注释掉，可正常使用



1. 已经实现用数据包控制上传，初步测试都没问题。明天再添加空包模式。
2. 去挑婚纱照啊。哎！最终还是被坑了多要了9张照片，450+150运费。。。不开心

# 2018/10/31

1. 今天打开了昨天调试好的设备，居然会出现乱七八糟的状况。还不知道是什么状况，有点担心.
2. 难道是手机蓝牙没关，一不小心脸上了？
3. 搞死了，刚好发送端spid =0x10 接收端pid=0x10 导致乱码输出。
4. 空包以及数据包都带有上传操作，出现了接收到的数据比实际数据要多的情况，初步猜测是空包上传把pid注释掉了，当收到重复包时，重新上传的操作。
5. 初步测试，命令包上传，以及数据包上传都没问题了。现在定好30000次，达到次数就停止。明天查看结果。

# 2018/11/1

1. 4:20左右起床，看看linux视频。6点多又要做早餐、做饭。我宝起来说我没放鸭蛋在饭里蒸，生我气，想想每天早上起来做饭，感觉都成为理所当然了。

可能也是我4点多起来有点生气吧。

1. 8点38左右到公司，查看测试记录，预定30000条，只有29674,正确的有29646次。也就是有28条乱码。初步测试也算是这样了，正常。需要排查问题，从数据上入手吧。
2. 考虑到了之前的想法行不通，需要换个方向。改变成接受器给发送端主动发送成功指令，之前自带的回应响应信息忽略。（关于crc校验的问题）
3. 下班时，把枪改成不发命令包，接收端把命令包添加的代码注释掉，一步步排查。
4. 发送端设置发送到8w次就停止，明天查看结果。

# 2018/11/2

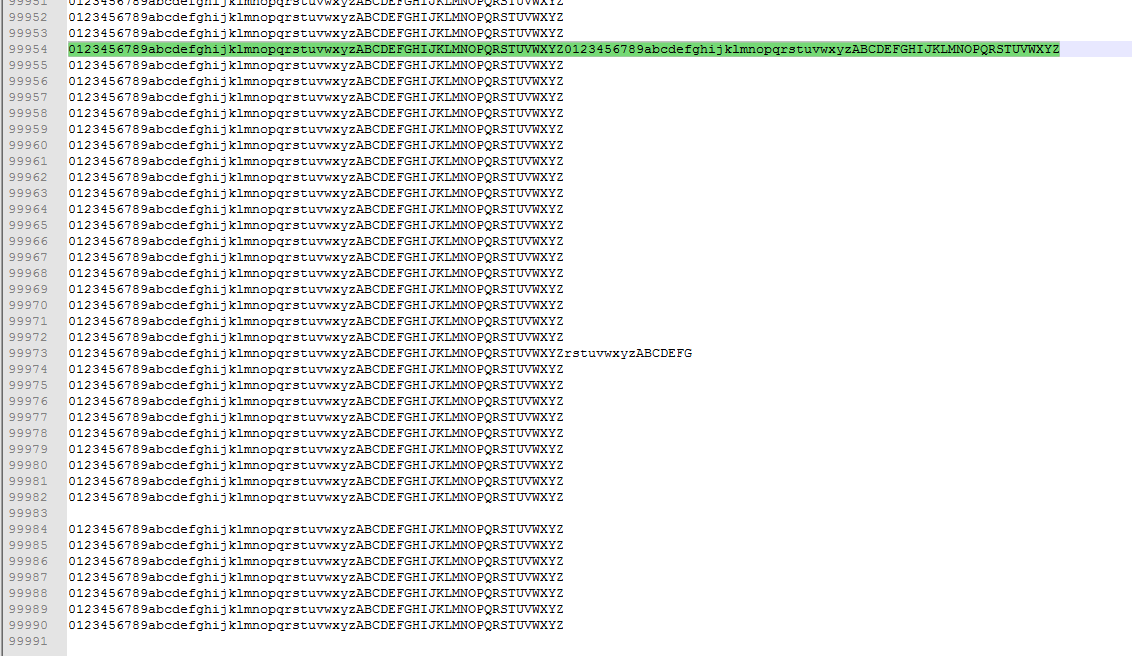
1. 锐雄问我一些事情，突然间想起新修改的方案也有这个问题。需要修改，测试。
2. 接收端需要在接收完整条数据后，发送一个确认pid，以防止发送方收到之前的响应，误操作。
3. 安装了虚拟机测试，但是响应较慢。不行
4. 测试接收器主动发送数据，成功。
5. 接收端主动发送数据时，效果太差，正在调试。
6. 裸机错误，接收端加了crc校验，发送端没加上，导致校验出错，所以没接收端没发送响应。目前效果还不错。

# 2018/11/5

1. 接收端主动回复收到的pid，发现成功率有点低，影响流畅性。经过多次试验，发现每次发送完成需要一定的延时，才能保证稳定性
2. 重新测试10000次，0-9a-zA-Z没问题。
3. 重新测试了几次500都没问题
4. 计划今晚测试10w次，间隔为0 ，只使用数据包上传数据。正常应该收到10w-1次。明天查看结果，估计明早9点多，才能到达10w次。

# 2018/11/6

1. 昨晚的测试结果，还真的很巧，刚好到99954条出错，发送10w条，显示效果：



仔细查看了下发送端的代码，成功了10w条，没什么问题，难道是接收端发生看门狗复位了？有可能。

1. 由于发送端修改的某些地方，导致配对不成功。后期需要修改。
2. 仍然有丢失，难道是USB上传速度太快？
3. 通知上传的命令接收出错，导致缺失？
4. 将uart.buff[0]改成第一个数据包的crc校验值，防止第一个数据出错时，将buf中的数据替换，导致出错。
5. USB上传测试，速度调慢，测试一晚上，看有没有出错
6. 将usb上传前，添加一个数据校测。如果与0123...不同，则打印error。表示数据发生错误。如果有error，表示数据出错。没有error但是有不正确的数据，则代表usb上传出了问题。

2018/11/7

1. Usb键盘有一部分是键值方式上传的，需要修改成编码输出方式，疏忽了
2. 在测试软件上，误码比较多。记事本表现较好。
3. 将测试方法，改成usb打印序号的方法，数据出错就打印error。这样就不用担心，输出数据出错时，不确定是2.4G问题，还是USB问题。明天如果没出现error，且数据达到23w-1，则代表2.4G没问题，可以进行下一步。
4. 已经测试20258次，没问题