Neodun 驱动与签名机通信协议

20170730：

上位机驱动 driver简称D

单片机签名机 signer简称S

之前的简称c s 比较临时，废弃不用

D2S 表示上位机发往单片机的报文

S2D 表示单片机发往上位机的报文

**报文通用设计**

报文长度 64字节

后文说第几字节，一律从零开始计数

第0 1字节 用作报文识别信息，byte1 byte2

第2 3字节 用作响应识别串号， 是一个小头在前的 uint16

第 62 63 字节 用作报文验证位，是对报文0~62字节做crc32运算之后的整数，以小头在前存储，去前两个字节。

第 4~61 共58个字节用作报文数据区，由每条报文独自约定

关于丢包，上位机向单片机发送，丢包是异常，可以捕获。

增加一些设计

第一字节为分类字节

第二字节小于a0的是普通报文，msgid 是随机数

第二字节大于等于a0的是回复类报文，回复类报文是针对某条报文的回复，msgid 和该报文一致

第二字节大于等于e0的是错误类报文，用来报告错误，错误报文都是回复类报文

回复类报文没有对应的回复类报文

后面要严格遵守这些设计，有不匹配的地方，要尽快调整

报文表达方式

D2S[tag1,tag2],tag用十六进制byte

比如D2S[01,10] ,如果有D2S 0x0110是以前的不规范表达，遇到请修改

**数据传输报文**

数据传输采取被动式设计，发送方发起传输指令，接收方自己根据需要获取，至获取完毕后，接收方发出确认指令。

数据传输类报文 第一字节都是01

异常通知和收到确认这样的报文，第一字节都是 00

第二字节大于等于a0的全部是回复类报文x

协议响应数据从 [01,a0]开始

协议错误响应数据从 [01,e0]开始

标准错误响应从 [00,e0]开始

1.D2S [01,01] 向签名机发送数据包

第2 第 3 字节，串号取随机数（除了是针对特定指令的确认和异常通知是取相等串号，其余全是随机数，后文不再赘述）。

第4 -7 字节 一个 小头在前的 uint32，表示传输数据大小

第8~39 字节，32个字节的 hash256 数据，表示传输数据的特征码

2.S2D 回复，串号都是 D2S0x0101的串号

S2D [00,e0] //报文出错，报文最后两个字节 和 报文内容验证不匹配

无数据，表示报文数据区全0，后文不再赘述

S2D [01,e0] //协议出错，数据容量太大，无法缓存

无数据

S2D [01,12] //数据接收完毕，但是hash256 不匹配

第4~7字节是一个uint32，表示对该数据的分配编号。

第8~39字节hash256

S2D [01,11] //数据接收完毕，并且hash256 匹配

第4~7字节是一个uint32，表示对该数据的分配编号。

第8~39字节hash256

传输完成之后，D2S 可以再发送其他控制指令，指定用制定编号的缓冲区数据进行操作。单片机上最多同时保持四个缓冲区数据（暂定）。

3.S2D请求数据

S2D [01,10] 请求数据，将原数据按照 50 byte 分片，最后一片不足50byte 补0

第4~5 字节是一个uint16，表示请求数据的开始片数

第6~7 字节是一个uint16，表示请求数据的结束片数

第8~39字节，hash256 数据，请求内容的特征码

4.D2S响应请求

D2S [01,a2] （回复协议）

2 3字节的串号是 S2D [01,10]的串号

4~5字节是一个uint16，表示发送哪一片的数据

6~55字节是数据

D2S 0x01a3（回复协议）

2 3字节的串号是 S2D 0x0110的串号

请求的数据全部发送完毕

5.S2D数据确认

收到[01,03]后，signer判断数据是否接收完整，不完整继续发送数据请求

完整了做判断后回复 0x01a0 或者0x01e1

数据确认的是报文D2S 0x0101，所以用D2S 0x0101 的串号

只有一条协议是另一条协议的回复协议，才用他的串号。

单片机向上位机发送数据

上述协议S2D 变成D2S，D2S变成S2D 即可

**签名相关报文**

签名相关报文，第一位都是 02

协议响应数据从 [02,a0]开始

协议错误响应数据从 [02,e0]开始

D2S [02,01] 查询设备地址情况//查

回复，回复类报文 2~3字节都是请求报文的串号

S2D 02a0 （返回一条地址）

4 5字节 是uint16 当前地址index，

6 7 字节是 uint16 总共有多少个地址

8 9是 地址类型 01 01 表示小蚁地址

10~61 是 hash格式地址，长度不够的地方填0

S2D 02a1 (返回地址结束)

4 5 字节是 uint16 总共有多少个地址

D2S [02,02] 设置地址信息//改

给地址设置点图标、昵称什么的

…..稍后补充

D2S [02,03] 删除地址以及所有信息（需要signer验证）//删

4 5 字节 是地址种类的标识

6~45 字节是预留四十个字节的地址，根据不同地址类型编码方式不一

S2D[02,c1] 成功

S2D[02,c2] 失败

D2S [02,04] 增加一个地址//增加

4 5 字节 是地址种类的标识

6~45 字节是预留四十个字节的地址，根据不同地址类型编码方式不一

//对neo来说，是 address字符串的 base58 解码bytes，不足填0

46~49 字节是参数0数据块id 暂时用私钥明码数据块，流程调通以后会加密

50~53 字节是参数1 数据块id ，0表示无此参数

54~57 字节是参数2 数据块id ，0表示无此参数

58~61 字节是参数3 数据块id ，0表示无此参数

回复

S2D[02,b1] 地址增加成功

4 5 字节 是地址种类的标识

6~45 字节是预留四十个字节的地址，根据不同地址类型编码方式不一

S2D[02,b2] 地址增加失败

4 5 字节 是地址种类的标识

6~45 字节是预留四十个字节的地址，根据不同地址类型编码方式不一

D2S [02,06] 获取地址安全信息（私钥等）（需要signer 验证） 疑问？

4 5 6 7固定填0

8 9 是地址类型，10~61是 hash格式地址，长度不够的地方填0

回复

S2D [02,a4] （返回地址数据块id）

第4 -7 字节 一个 小头在前的 uint32，表示传输数据大小

第8~39 字节，32个字节的 hash256 数据，表示传输数据的特征码

然后上位机可以用D2S 0x0110，分片将安全数据要回去

S2D [02,e0] signer拒绝

D2S [02,08] 设置权限

给地址设置加密方式什么的

…..稍后补充

D2S [02,0a] 对指定数据块进行签名 疑问：源码是0101

4 5 字节 是 签名方法 00 01 表示neo交易签名，以后可能会有其他

上位机发送数据后会收到四字节的数据编号，S2D 0x01a0 发给上位机的

6 7 8 9 是编号0

10 11 12 13 是编号1

14 15 16 17 是编号2

18 19 20 21 是编号3

….

不同的签名方法可能需要不同数量的参数

每个编号代表之前发送来的数据块

回复

S2D 02a8 (返回数据块id)

第4 -7 字节 一个 小头在前的 uint32，表示传输数据大小

第8~39 字节，32个字节的 hash256 数据，表示传输数据的特征码

然后上位机可以用D2S 0x0110，分片将安全数据要回去

S2D 02e3 signer拒绝

D2S [02,0b] 安装更新app文件

4 5 字节 uint16 表示密码的长度（目前固定6位）

6~61是 密码（DecodeBase58 6个数字组成的字符串）

回复

S2D 02c3 成功

S2D 02c4 失败

D2S [02,0c] 验证钱包密码

4 5 字节 uint16 表示密码的长度（目前固定6位）

6 7 字节是告知要开启的功能 byte1 byte2 例如增加地址就是02 04

8~61是 （DecodeBase58 6个数字组成的字符串）

回复

S2D 02c5成功

S2D 02c6 失败

D2S [02,0d] 获取钱包是不是新钱包（废弃）

回复

S2D 02b5 是

S2D 02b6 不是

D2S [02,1b] 获取配置信息（1true0false) 签名机初始时全部默认true

1. 5字节 是否是新设备 （UINT16）

6 7连接钱包后是否自动弹出驱动界面（功能暂缺，先false）

8 9开机时是否自动检查更新（功能暂缺，先false）

10 11新增地址时是否要密码验证

12 13删除地址是否要密码验证

14 15备份地址是否要密码验证

16 17备份钱包时进行是否要加密（功能暂缺，先false）

回复

S2D 02d1成功

S2D 02d2 失败

D2S [02,1a] 设置配置信息

同上 4 5字节不用处理

回复

S2D 02d3成功 回复同设置

S2D 02d4 失败

D2S [02,1c] 要求下位机显示密码验证页面并开启对应功能的验证功能

S2D 02d5成功

S2D 02e2 失败