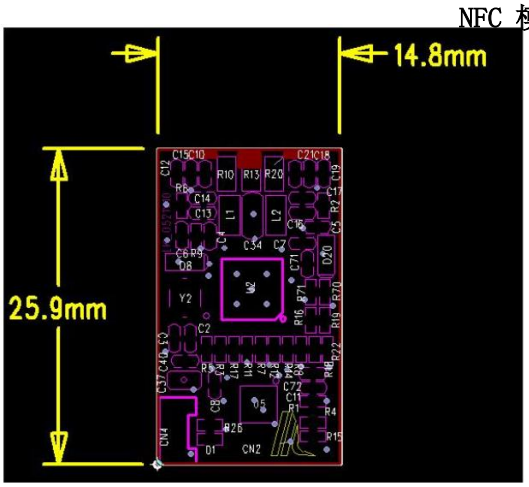
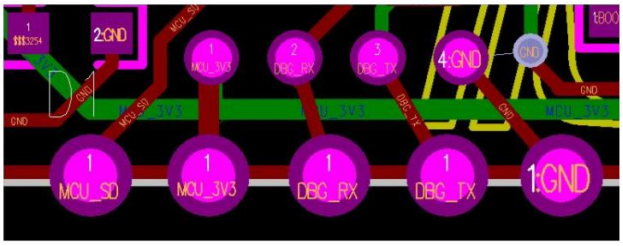
NFC 模块通信协议





本模块支持 LPCD 和用户控制模式寻卡，

UART 串口通信，支持波特率更改；
存储用户 刷卡认证数据；
自动读写块操作
支持 A,B 卡 读写
支持卡唤醒操作；

应用范围： 物联网，酒店，门锁，物业，电梯，电车，需要卡认证的行业；

以下操作是针对 Mifare 卡的操作

数据结构
本协议数据采用大端格式，内容格式视数据需求按内容要求

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	N	1
内容	0xFF 0x55	如列表	数据长度	有效数据	0-所有数据累加和

L=N+1
C=0-（T+L+V） 取 1 字节

```
typedef enum
{
    //cmd
    LK_CMD_ACCREDIT=0x11,//授权模式
    LK_CMD_UNCREDIT =0x22,//退出授权
    LK_CMD_RECREDIT    =0x33,//清除授权


    LK_CMD_XKEY    =0x40,//设置认证密钥
    LK_CMD_MREAD    =0x41,//设置读写卡模式
    LK_CMD_VER=0x53,//版本信息
    LK_CMD_TYPE=0x54,//设置属性
    LK_CMD_EXT=0x50,//扩展命令
    LK_CMD_AUTH=0x51,//认证
    //status
    ST_SUCCESS=1,
    ST_FAIL=2,
    }LK_CMD_e;
//=============================================
操作模式
#define SHORT_READ          0    //短读
#define LONG_READ           1    //长读
#define SET_READ            2    //设置
#define RET_NULL            3    //表示后面没数据
#define GET_DATA            4    //获取
//=============================================
数据性质
#define GET_READ_UID        3    //  卡 UID
#define GET_READ_NUM        4    //  银行卡号
#define GET_READ_IPH        5    // iPHone 卡号
#define GET_READ_CID        6    //  身份证号
#define GET_READ_INFO       7    //  注册信息
```

1:LK_CMD_ACCREDIT

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x11	00 02	n	0-（T+L+V）

该指令添加卡片，可以连续添加，最多 100 张

NFC 反馈 V：状态+操作模式+数据性质+数据长度+数据

状态：成功 1，失败 2

操作模式：0 短读；1 长读；2 设置

数据性质 :3 UID,4 NUM,5 IPH,6 CID
卡 ID 长度 ： 通常 5BYTE ,IPHONE 手机 卡 是 22 位，折合 11 字节，一般银行卡 8—16 位；
UID： 唯一卡识别号

2:LK_CMD_UNCREDIT

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x22	00 02	n	0-（T+L+V）

退出添加卡片，自动进入卡片比对模式
NFC 反馈 V：状态+操作模式+数据性质+数据长度+ 数据
状态： 成功 1，失败 2
操作模式：2 设置
数据性质 :7 注册信息
数据长度：2+2
数据： 有效用户数+用户总数

3:LK_CMD_RECREDIT

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x33	00 02	n	0-（T+L+V）

该指令 将原来存储的卡片 数据全部清除；
NFC 反馈 V：状态+操作模式
状态： 成功 1，失败 2
模式：

4:LK_CMD_XKEY

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x40	00 02	n	0-（T+L+V）

设置加密卡读卡密钥

5:LK_CMD_MREAD

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x41	00 02	n	0-（T+L+V）

读写卡片指定数据区；
NFC 反馈 V：状态+操作模式+数据性质+数据长度+　数据
状态：　成功　1，失败　2
操作模式：2　设置
数据性质　:3 UID,4 NUM,5 IPH,6 CID，

6:LK_CMD_VER

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x53	00 02	n	0-（T+L+V）

读取设备版本信息

NFC 反馈 V：状态+操作模式+数据性质+数据长度+　数据
状态：　成功　1，失败　2
操作模式：0,短读,1　长读,2　设置
数据性质　:3 UID,4 NUM,5 IPH,6 CID

7:LK_CMD_TYPE

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x54	00 02	n	0-（T+L+V）

未定义

8:LK_CMD_EXT

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x50	00 02	n	0-（T+L+V）

未定义

9:LK_CMD_AUTH

	同步码 S	指令码 T	长度 L	内容 V	校验 C
字节数	2	1	2	8/12	1
内容	0xFF 0x55	0x51	00 02	n	0-（T+L+V）

未定义

应用举例：

```
uint8_t Picc_PacketToApp(uint8_t cmd,uint16_t len,uint8_t *dat)
{
    uint16_t i;
```

```
    uint8_t sum;


    sum=0;
    //head
    SendToApp(0xFF);
    SendToApp(0x55);
    //T
SendToApp(cmd);          sum -=cmd;
    //L
    SendToApp((len+1)/256); sum -=(len+1)/256;
    SendToApp((len+1)%256); sum -=(len+1)%256;
    //V
    for(i=0;i<(len);i++)
    {
        SendToApp(dat[i]); sum -=(dat[i]);
    }
    //C
    SendToApp(sum);
    //start send
    StartupUartTx(APP_UARTx);
    return 1;
}

//授权模式
{0xFF,0x55,0x11,0x00,0x00,0xEF},
//退出授权
{0xFF,0x55,0x22,0x00,0x00,0xDE},
//清除授权数据
{0xFF,0x55,0x33,0x00,0x00,0xCD},
//设置认证密钥
{0xFF,0x55,0x40,0x00,0x12,.....0xNN},
//设置读卡模式
{0xFF,0x55,0x41,0x00,0xNN,.....0xNN},
//版本信息
{0xFF,0x55,0x53,0x00,0x00,0xAD},
//扩展命令
{0xFF,0x55,0x50,0x00,0xNN,.....0xNN},
//认证
{0xFF,0x55,0x51,0x00,0x00,0xAF}
```