昵称：阿玛尼迪迪
园龄：3年1个月
粉丝：63
关注：6
+加关注

< 2018年7月 >

日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

python(34)
opencv(10)
stm32(5)
支持向量机(3)
最短路径(2)
朴素贝叶斯(2)
决策树(2)
排序(2)
优先队列(2)
栈(2)
更多

随笔分类

ActiveMQ(3)
C++(87)
Caffe(12)
Java Web(2)
java菜鸟(4)
Leveldb(1)
Linux(18)
OpenCv(17)
oracle&sql(8)
python(36)
Stanford大学机器学习公开课(5)
stm32学习笔记(15)
vs2013(7)
编程思想(8)
分布式计算
国密(1)
机器学习(11)
计算机菜鸟(8)
剑指Offer(7)
设计模式(7)
数据结构(9)
算法(17)
图像处理(2)
网络编程(1)

随笔档案

2018年6月 (3)
2018年5月 (4)
2017年11月 (7)
2017年10月 (9)
2017年9月 (4)
2017年8月 (26)

stm32——NFC芯片--PN532的使用

一、NFC简介

NFC(Near Field Communication)近场通信，是一种短距高频的无线电技术，在13.56MHz频率运行于20厘米距离内。其传输速度有106Kbit/秒、212Kbit/秒或者424Kbit/秒三种。目前近场通信已通过成为ISO/IEC IS 18092国际标准、ECMA-340标准与ETSI TS 102 190标准。

NFC近场通信技术是由非接触式射频识别（RFID）及互联互通技术整合演变而来，在单一芯片上结合感应式读卡器、感应式卡片和点对点的功能，能在短距离内与兼容设备进行识别和数据交换。工作频率为13.56MHz.但是使用这种手机支付方案的用户必须更换特制的手机。目前这项技术在日韩被广泛应用。手机用户凭着配置了支付功能的手机就可以行遍全国：他们的手机可以用作机场登机验证、大厦的门禁钥匙、交通一卡通、信用卡、支付卡等等。

二、PN532概述

PN532是一个高度集成的非接触读写芯片，它包含80C51微控制器内核，集成了13.56MHz下的各种主动/被动式非接触通信方法和协议。

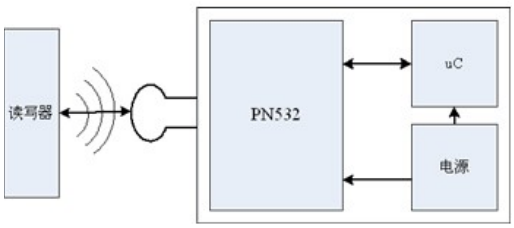
PN532传输模块支持6种不同的工作模式：

读写器模式，支持ISO/IEC 14443A / MIFARE®机制
读写器模式，支持 FeliCa机制
读写器模式，支持ISO/IEC 14443B机制
卡操作模式，支持ISO 14443A / MIFARE®机制
卡操作模式，FeliCa机制
ISO/IEC18092，ECM340点对点

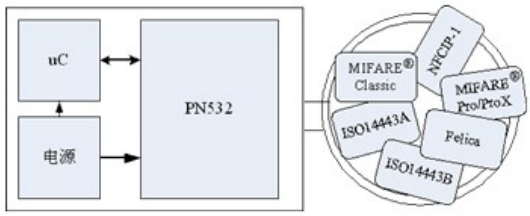
支持主机接口：

- SPI接口
- I^2C接口
- 串行UART

PN532的操作形式如下图所示：



卡操作模式



读写器模式

三、PN532模块搭配ISO/IEC14443Type A-4卡片

我使用的是PN532模块搭配ISO/IEC14443Type A-4卡，使用串口模式。这个卡是使用APDU(Application Protocol Data Unit--应用协议数据单元)的指令形式操作的。我拿到的卡片是由发卡商提供的，发卡指的是卡的初始化，即卡号写入和密钥写入的过程，完成后卡片才可发放出去；

发卡：

- 注入卡号，只能注入一次；

2017年7月 (24)
2017年4月 (7)
2017年3月 (4)
2017年2月 (4)
2017年1月 (13)
2016年12月 (19)
2016年11月 (7)
2016年10月 (1)
2016年9月 (30)
2016年8月 (33)
2016年7月 (10)
2016年6月 (11)
2016年5月 (13)
2016年4月 (25)
2015年11月 (1)
2015年10月 (1)

最新评论

1. Re:积分图及其应用
博主, 积分图的好文, 学习了
--Just_ForFun
2. Re:C++多线程编程（入门实例）
讲的不错哈
--搞笑科技
3. Re:Adaboost算法结合Haar-like特征
您好，谢谢分享。我有疑问，文中说特征大小为2*1的应该是图一的1（a）吧，而不是图二。
--woaixuexi
4. Re:C++多线程编程（入门实例）
为什么在主程序中建立了线程又在主线程执行前关闭了线程，而且程序能够正确运行，多线程小白，求解答
--文月
5. Re:静态库和动态库的区别
简单易懂，一直搞不清各种lib.a和各种lib.so区别，看了大神的文章茅塞顿开，多谢大神分享！
--不想脑子空空

阅读排行榜

1. C++多线程编程（入门实例）(55074)
2. C++文件读写详解（ofstream,ifstream,fstream）(11344)
3. 题目：在一个二维数组中，每一行都按照从左到右递增的顺序排序，每一列都按照从上到下递增的顺序排序。请完成一个函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数。(9950)
4. Linux中error while loading shared libraries错误解决办法(9083)
5. stm32——RTC实时时钟(8822)

评论排行榜

1. C++多线程编程（入门实例）(4)
2. 值传递和引用传递-----函数参数传递的两种方式(1)
3. 题目：在一个二维数组中，每一行都按照从左到右递增的顺序排序，每一列都按照从上到下递增的顺序排序。请完成一个函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数。(1)
4. 积分图及其应用(1)
5. 静态库和动态库的区别(1)

推荐排行榜

1. C++多线程编程（入门实例）(10)
2. 题目：输入一个链表，从尾到头打印链表每个节点的值(3)
3. stm32——RTC实时时钟(2)
4. libsvm的数据格式及制作(2)
5. 【转】回调的原理、应用(1)

- 注入密钥，各个密钥只能注入一次；

接口调用：

```
S> C0 F5 00 01 08620000000000001F //卡号
<R 90 00
S> C0 F5 00 02 10701279D95F77B378C735F17A019EFA6E //外部认证密钥
<R 90 00
S> C0 F5 00 03 103A748687BFA62A808B4C87AF0EE4B468 //充值密钥
<R 90 00
S> C0 F5 00 04 1040CFE895E6076C932422C04F448C0CA1 //充值通讯密钥
<R 90 00
S> C0 F5 00 05 10C501CBE8A849B3E7F638E7E096E560EF //消费密钥
<R 90 00
S> C0 F5 00 06 104D2A98A9165C79F5A54C862A367E4969 //存储密钥
<R 90 00
```

我使用该卡的目的是做一个刷卡饮水的系统，其中涉及卡片的步骤为：激活----寻卡-----读卡号----写入外部认证----刷卡消费

1、激活

PN532自带一个休眠功能，要使用PN532对NFC卡片进行读写的时候要激活一下（唤醒），一般放在程序的开头，调用一次即可。

激活PN532发送的命令（十六进制）为：

```
-> 55 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 03 FD D4 14 01 17 00
```

线没接错的话就会返回：

```
<D 00 00 FF 00 FF 00 00 00 FF 02 FE D5 15 16 00
```

2、寻卡

寻卡是涉及到卡片的第一个步骤，nfc对卡的操作都是先寻找卡的，若寻不到卡，则后续的读写操作将无法进行；反之，若寻到卡，则后续的读写操作过程中将不再对卡片进行身份确认。

寻卡命令：

```
-> 00 00 FF 04 FC D4 4A 01 00 E1 00
```

上面的数据中：

00 00 FF----帧头，与卡相关的所有命令都要包含该头；

04----包长度，即“D4 4A 01 00”的长度；

FC----包长度校验：0x100-包长度；此处0xFC=0x100-0x04;

D4----方向标识码：数据方向是stm32（主机）至PN532

4A----寻卡命令标识码

01----卡数量，最大是2，PN532一次只能同时处理两张卡

00----PN532工作模式：106 kbps type A (ISO/IEC14443 Type A)

E1----发送的数据校验：等于：0x100-(0xff&(0xD4+0x4A+0x01+0x00))

00----暂时没发现有什么卵用；

对于我的卡，若寻到卡，则返回：

```
<D 00 00 FF 00 FF 00 00 00 FF 20 E0 D5 4B 01 01 00 84 20 0A 01 23 45 67 89 AB CD EF
AA 55 0E 78 77 84 03 66 52 57 45 32 50 52 4F 4D 29 00
```

返回的命令也是有规则可循的，但你看上面太长了，对返回的命令部分，在下面再讨论吧（下面短）。

3、读卡号

接口调用：

```
S> C0 F1 00 01 08
<R 62 00 00 00 00 00 00 1F 90 00
```

所以读卡号的命令是：

```
->00 00 ff 08 f8 d4 40 01 c0 f1 00 01 08 31 00
```

看吧：00 00 ff----帧头；08----包长度（d4 40 01 c0 f1 00 01 08）；f8----0x100-0x08；d4----主控至PN532；40----数据读写命令；01----卡1；c0 f1 00 01 08----读卡号命令；31----数据校验：等于：0x100-(0xff&(0xd4+0x40+0x01+0xc0+0xf1+0x00+0x01+0x08))

我的卡，返回的命令码是：

```
<D 00 00 FF 00 FF 00 00 00 FF 0D F3 D5 41 00 62 00 00 00 00 00 00 1F 90 00 D9 00
```

现在短了，可以分析一下啦：

00 00 FF 00 FF 00----ACK（应答）指令码：千万别以为收到应答码就以为卡片正常读写了，因为不正常读写也会收到应答码哟。原因是你操作PN532芯片，只要芯片工作正常，都会有应答码的啦。所以还得往下看，看后边的命令码是否符合规则。

00 00 FF----还记得吗？帧头----与卡相关的所有命令都要包含该头，所以无论是发送还是接收，都要包含帧头的。

0D----包长度，长度13（从D5到90 00正好是13）

F3----包长度校验

D5----方向标识码：PN532至主控

41 00----读正确标志位，若为其它如41 17之类的数据，就要注意了哦，说明读出错辣。

62 00 00 00 00 00 00 1F----卡号；卡号为8字节BCD编码，最大卡号支持16位数字。卡号小于16位数字时填充F。

90 00----APDU命令正确时的通用码子（错误时为69 00）；

D9----返回的数据校验：等于：0xff-0xff&(SUM(00 00 FF 0D F3 D5 41 00 62 00 00 00 00 00 00 1F 90 00))

4、外部认证

为了安全考虑，卡片需要进行外部认证。其步骤如下：

- 外部设备从卡片获取16字节随机数；
- 外部设备使用“外部认证密钥”对随机数使用SM4_ECB方式加密；
- 外部设备将加密结果送入卡中，卡片对其进行解密，并对比随机数与解密结果是否一致；
- 卡片返回认证结果。

接口调用：

```
S> C0 84 00 00 10 //读取随机数
<R 1B 93 C6 32 91 A3 65 8B 2E D7 5D 90 49 B9 34 4F 90 00
S> C0 82 00 00 10EC15BF495C66D569C654C02AFBCAA3A2 //写入加密结果指令
<R 90 00 //验证成功
```

所以主控首先要获取卡片的随机数：（命令码解析规则和前述一样，在此不再赘述啦）

```
->00 00 ff 08 f8 d4 40 01 c0 84 00 00 10 97 00
```

返回：

```
<D 00 00 FF 00 FF 00 00 00 FF 15 EB D5 41 00 15 5E 1E A0 04 77 67 25 4B 0B E4 9B 7D
60 21 05 90 00 4A 00
```

分析得到卡片发来的随机数为：

15 5E 1E A0 04 77 67 25 4B 0B E4 9B 7D 60 21 05 （当然，每次获取的数都不会相同）

接着主控获取上面那串16字节随机数，使用已经写好的SM4加密算法对其进行加密。加密后的数据如下：

A5 96 7E 70 C3 B8 55 9E BB A7 E4 BA 12 95 7D 2E

然后外部设备将加密结果送入卡中，写入加密结果的指令为：

```
->00 00 ff 18 e8 d4 40 01 c0 82 00 00 10 A5 96 7E 70 C3 B8 55 9E BB A7 E4 BA 12 95 7
D 2E B0 00
```

返回：

```
<R 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF FB 41 00 90 00 5A 00
```

看到90 00就可以安心了，外部认证成功！

5、刷卡消费

出于安全考虑，刷卡消费过程还需要使用消费密钥进行数据的加解密，步骤如下：

- 外部设备从卡片获取14字节随机数
- 外部设备使用“消费密钥”加密14字节随机数+2字节消费金额，组成16字节消费报文；
- 外部设备将加密结果发送至卡片；
- 卡片解密消费报文，判断消费报文是否正确，执行消费，并拼接“DONE”+2字节消费金额+“AMOUNT”+4字节余额，用消费密钥加密后返回给外部设备；
- 外部设备调用解密算法解密收到的报文，并判断对比是否为““DONE”+2字节消费金额+“AMOUNT”+4字节余额”的格式，若正确，则开水阀打开。

接口调用：

```
S> C0 84 00 00 0e
<R 37 F3 10 E1 ED CC 07 F9 34 4B 39 50 1A 82 90 00
S> C0 85 00 00 1036E9DEBBFB21A6C6482568EAEA791ECF
<R F1 ED 7F 4C D5 1E AA D5 35 E6 62 63 6E 89 12 5D 90 00
余额:654.60
```

所以主控首先要获取卡片的14字节随机数，命令如下：

```
->00 00 ff 08 f8 d4 40 01 c0 84 00 00 0e 99 00
```

返回：

```
<D 00 00 FF 00 FF 00 00 00 FF 13 ED D5 41 00 37 F3 10 E1 ED CC 07 F9 34 4B 39 50 1A
82 90 00 E2 00
```

分析得到卡片发来的14字节随机数为：

37 F3 10 E1 ED CC 07 F9 34 4B 39 50 1A 82

再加上两字节的消费金额，这里我假定消费了1分钱，即两字节为:0x00 0x01

所以外部设备使用消费密钥，调用SM4算法对：37 F3 10 E1 ED CC 07 F9 34 4B 39 50 1A 82 00 01 进行加密，加密后的数据如下：

36 E9 DE BB FB 21 A6 C6 48 25 68 EA EA 79 1E CF

然后外部设备将加密结果发送至卡片，命令如下：

```
->00 00 FF 18 E8 D4 40 01 C0 85 00 00 10 36 E9 DE BB FB 21 A6 C6 48 25 68 EA EA 79 1
E CF 47 00
```

返回：

```
<D 00 00 FF 00 FF 00 00 00 FF 15 EB D5 41 00 F1 ED 7F 4C D5 1E AA D5 35 E6 62 63 6E
89 12 5D 90 00 F9 00
```

即卡片拼接“DONE”+2字节消费金额+“AMOUNT”+4字节余额，并用消费密钥加密后的发送给外部设备的数据为：

F1 ED 7F 4C D5 1E AA D5 35 E6 62 63 6E 89 12 5D

最后外部设备再对该串密文使用消费密钥，调用SM4解密算法进行解密，得到：

44 4f 4e 45 00 01 41 4d 4f 55 4e 54 00 00 ff b4

其中：

44 4f 4e 45----DONE（ASCII码的十六进制正好为44 4f 4e 45）；

00 01----2字节消费金额，即一分钱

41 4d 4f 55 4e 54----AMOUNT（ASCII码的十六进制正好为41 4d 4f 55 4e 54）

00 00 ff b4----4字节余额，即余额为654.60元

それでも私の大好きな人

分类: stm32学习笔记

标签: stm32, NFC

阿玛尼迪迪
关注 - 6
粉丝 - 63
+加关注

1 0

« 上一篇：python基础——单元测试
» 下一篇：题目：输入某二叉树的前序遍历和中序遍历的结果，请重建出该二叉树

posted @ 2016-09-06 21:42 阿玛尼迪迪 阅读(8123) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部