# 配对器

## 硬件

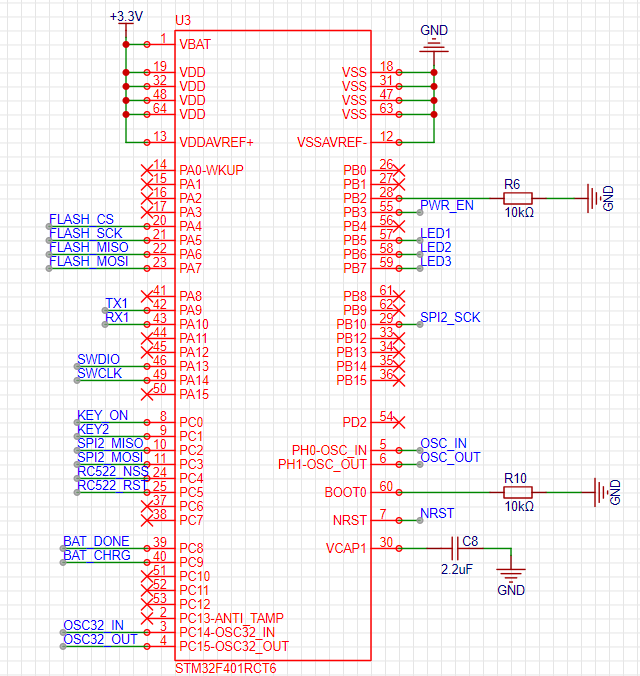
### 电路

电路嘉立创EDA，PCB项目名：读卡器V1.0

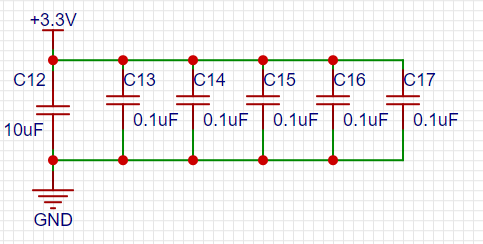
配对器电路由以下部分组成：

#### Stm32f401rct6单片机电路：

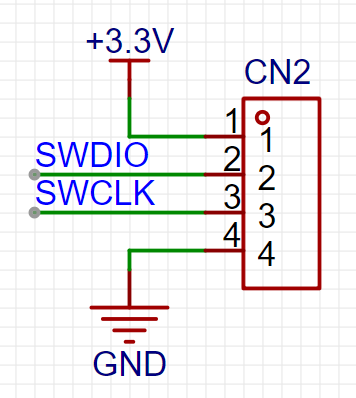
单片机主体：



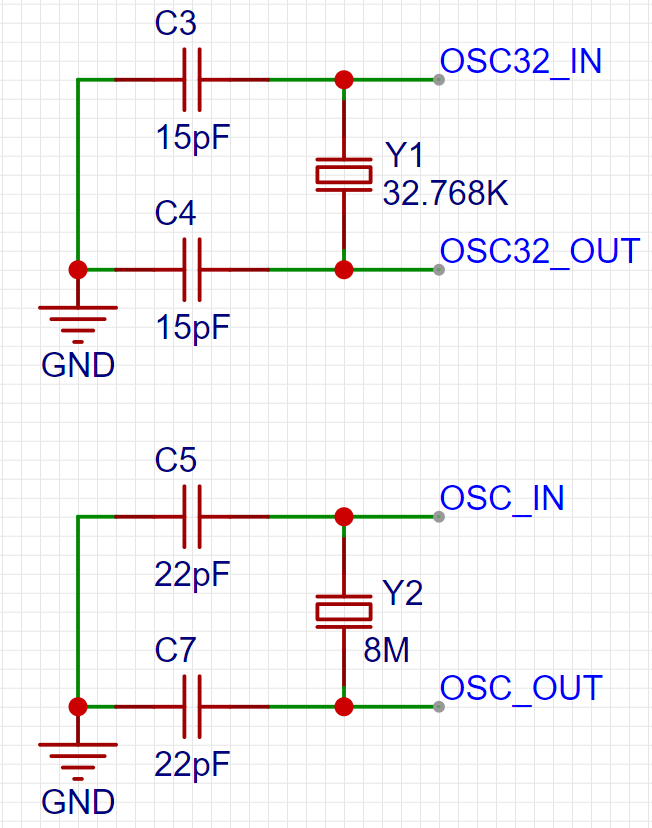
单片机的滤波电容（这些电容摆在单片机芯片的各个VDD口）：



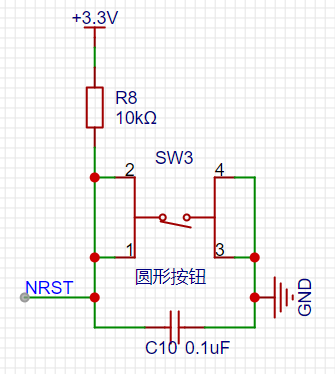
烧录入口：



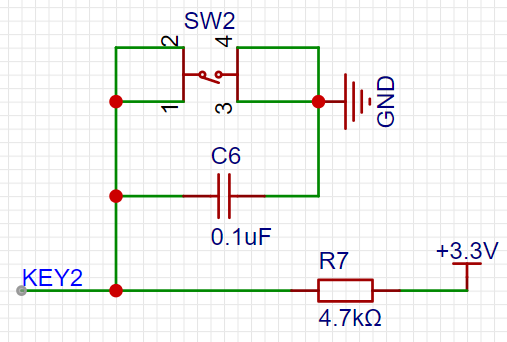
晶振：



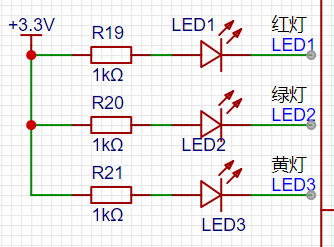
复位按键：



单片机按键：

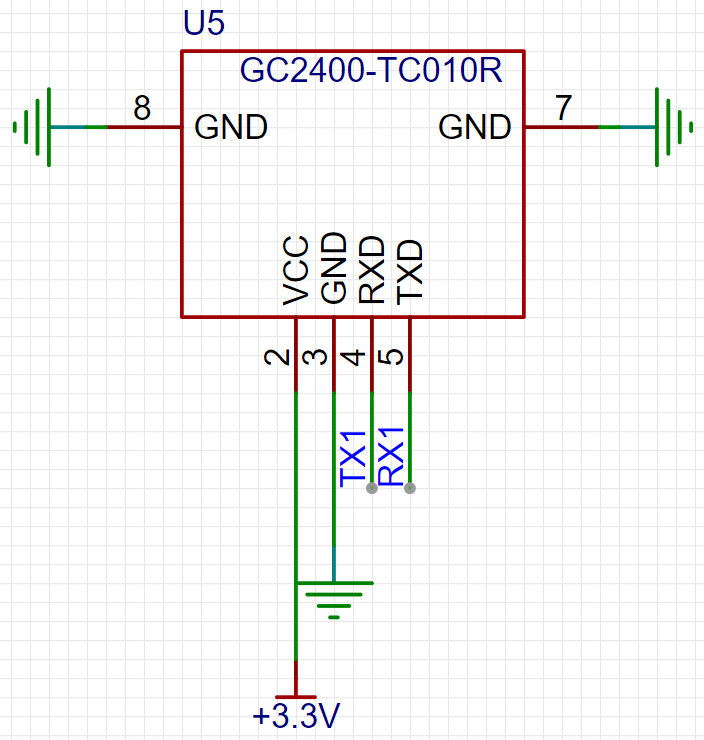


三色灯：

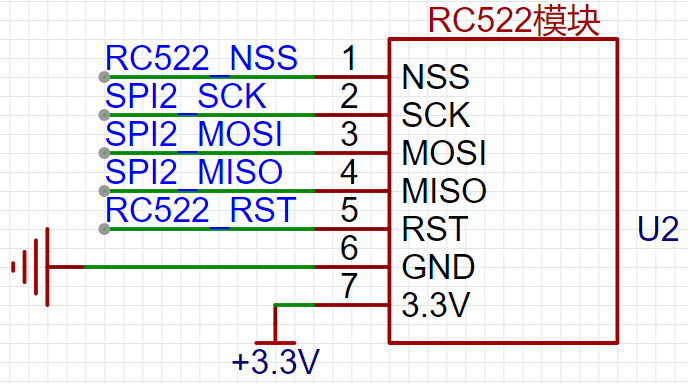


#### 2）外接模块电路：

2.4G无线模块：

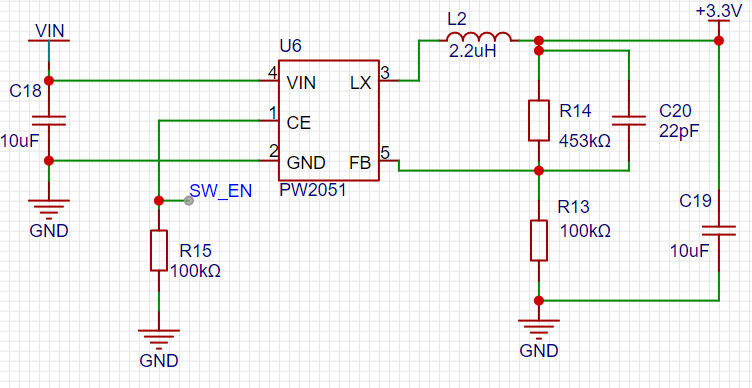


RC522模块：

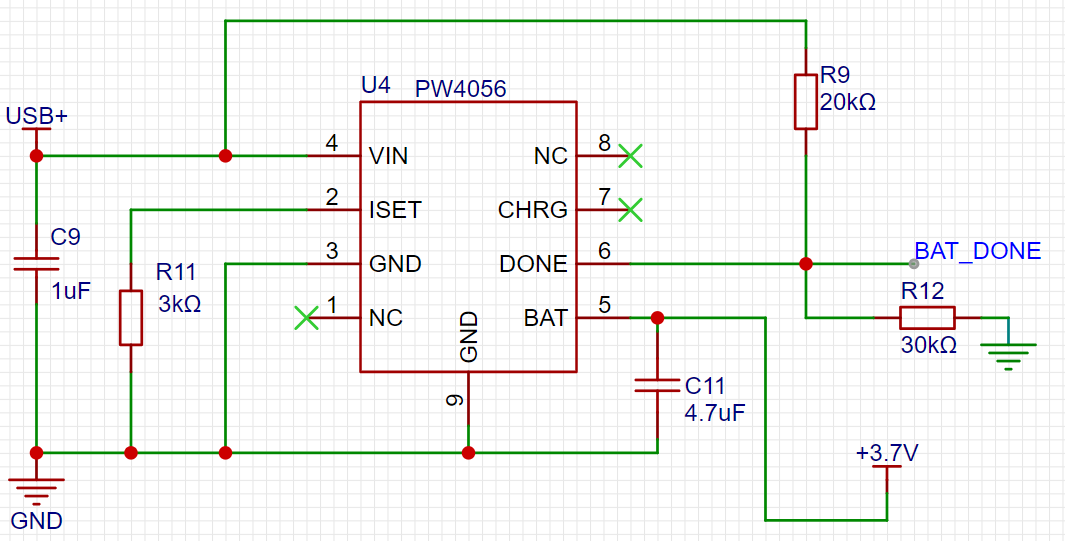


#### 3）供电电路：

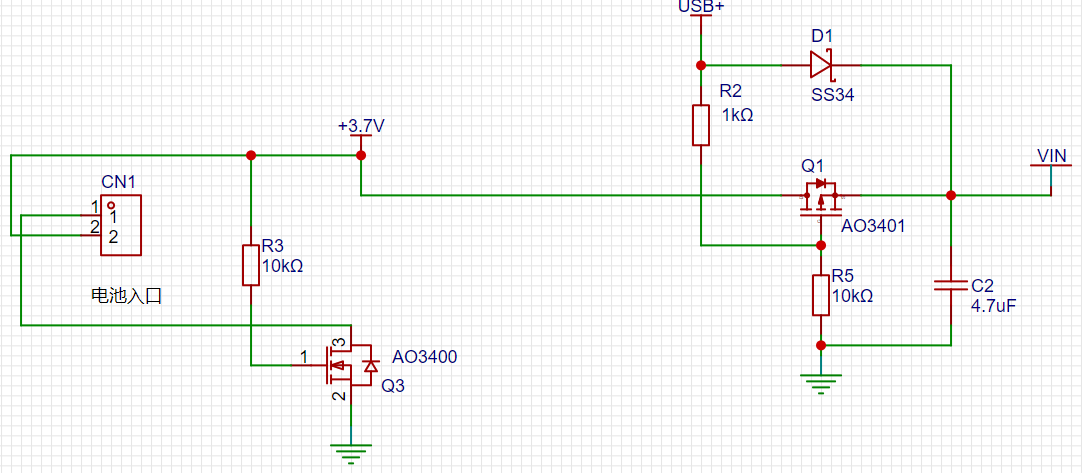
PW2051芯片的3.7V转3.3V电路：



PW4056电池充电电路：

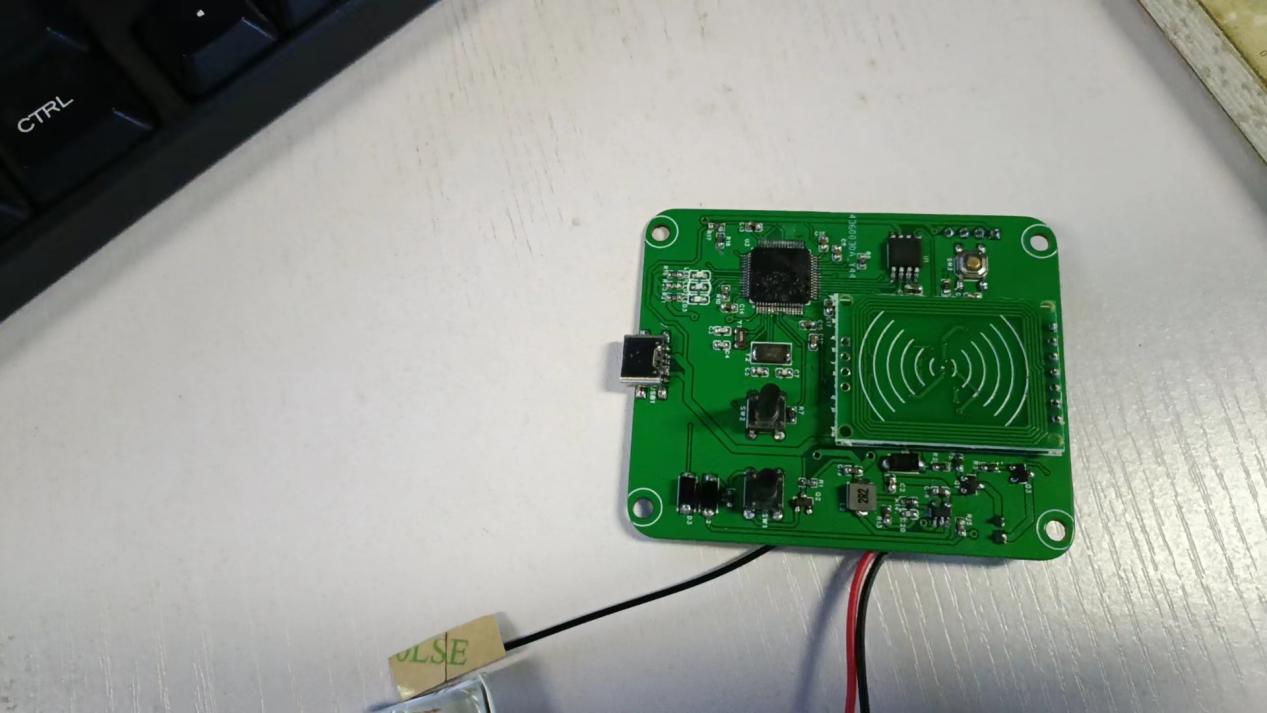


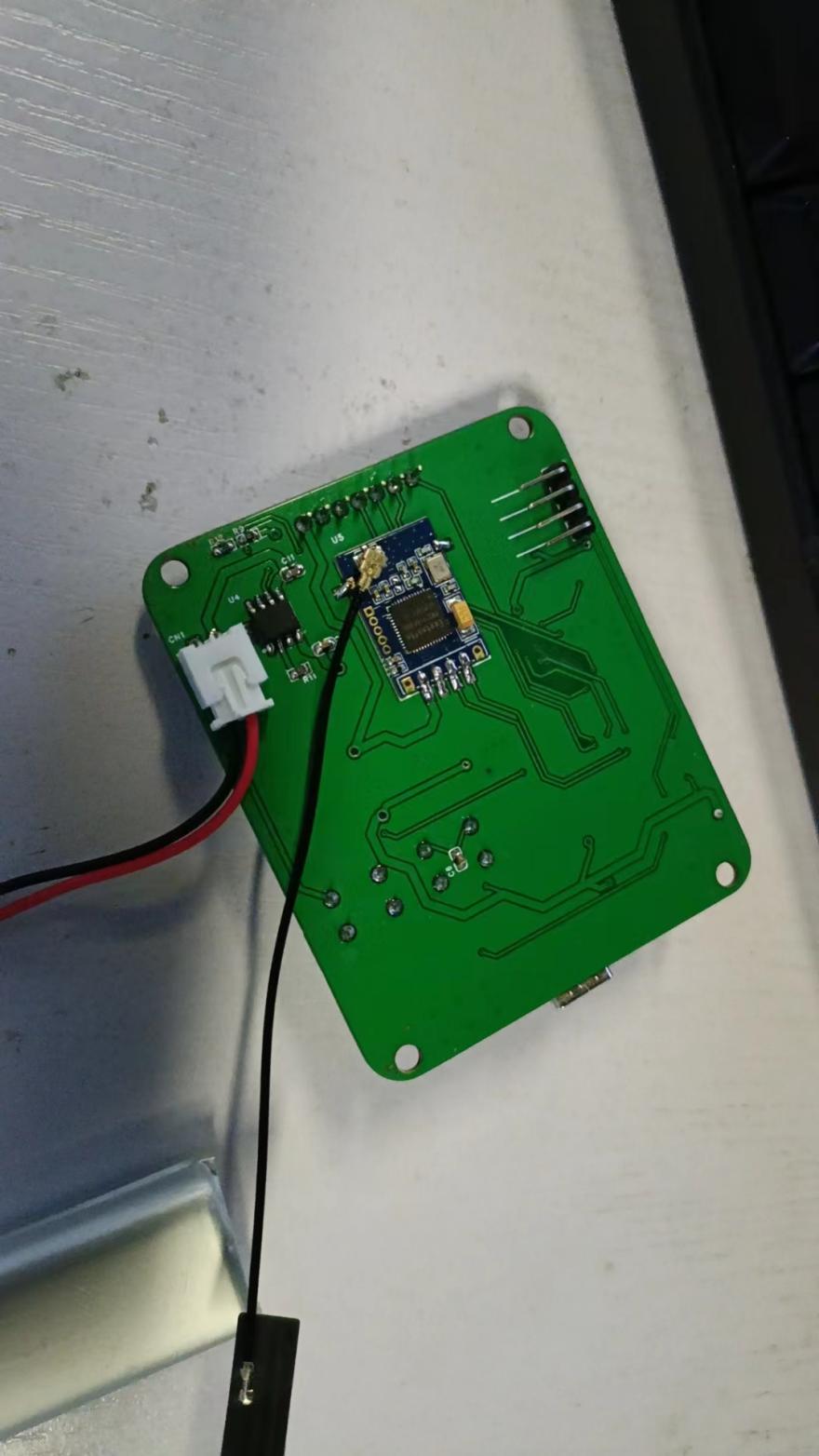
供电电源电路，有防反接功能：



### 实物与选型

#### 1）主板





#### MFRC522模块

选择MFRC522-mini版射频NFC模块。



#### 电池

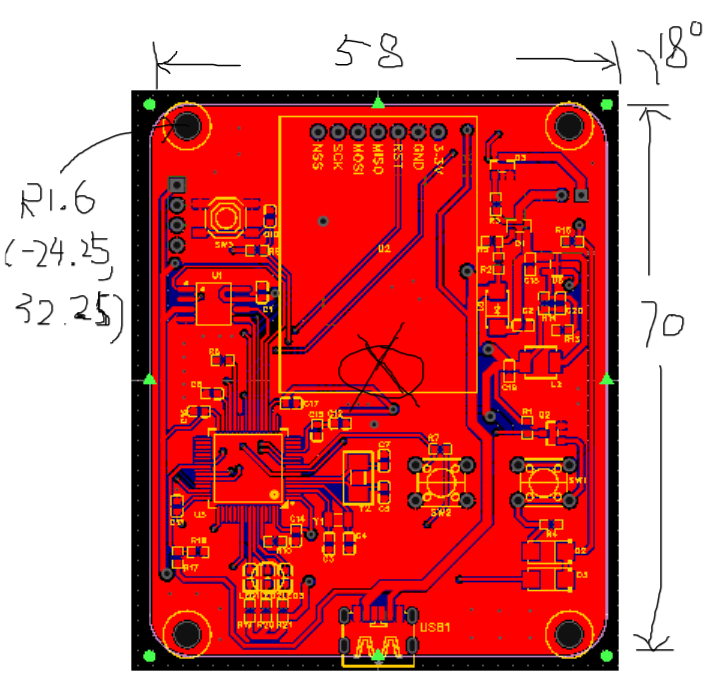
选择和枪一样的1400mAh电池。



#### 外盒



### 电路板尺寸



## 功能描述

使用S50卡与读卡器进行接触，读卡器读取到卡片中的信息。卡片中的信息是枪的IP地址信息。读卡器读取到信息后，发送信息给靶，并更改自己地址，最终达到修改地址与枪地址匹配的功能。

读卡器有三种色灯，绿色为“正常状态”。当绿灯状态时接触到S50卡，并成功读取到卡后，灯会变化为黄色，此时开始与靶进行无线传输。若成功，灯回复绿色。若失败，则会闪红灯一下，接着再变回绿色。

读卡器有两个按钮，右边按钮为长按开关机，左边按钮为操作键。按下左按钮，此时读卡器由绿灯变黄灯，而与该读卡器匹配的靶也会亮灯，此时读卡器进入“待修改状态”，注意此时无法读S50卡了。1）在该状态下可由安卓APP控制靶发送地址给读卡器，修改读卡器地址。2）在该黄灯状态下若长按右按钮，则灯会变为红色，再连按8次左按钮，红色闪烁后变为绿灯，此时读卡器的IP地址会重置为0x0000。

充电时读卡器会亮红灯，此时无法读卡也无法进行修改。

## 软件

### 卡

卡选择钱币卡，注意卡片要买S50或复旦芯片的卡片。使用其他芯片的卡要改程序。

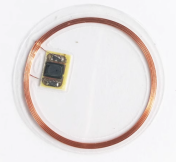
卡的程序需要实现写卡功能，把卡片写入信息。

写卡可单独找一个没用到的读卡器板，或用黑色F411开发板也可以。

#### 读卡：

读卡器通过天线往空中发射射频信号，产生电磁波磁场。卡片靠近读卡器进入磁场，产生感应电流激活卡片中的芯片，卡片中的芯片再通过卡片里的天线往读卡器发射射频信号，便达到了信息传递的功能。

读卡器的板载天线回路： 

卡片藏内部的芯片和天线： 

#### 卡种类：

这种卡统称IC/ID卡，IC卡的意思是可以改写的，ID卡则不可以改写只能读，我们使用IC卡。IC卡片里的芯片不同来区分卡的种类，有S50、S70、UltraLight、PRO等。

项目中使用S50卡。S70卡和S50卡一样，S70存储容量更大些，这两统称M1卡。复旦卡是采用复旦国产芯片替代S50芯片，因此可替换S50卡。

#### S50卡：

S50卡有16个扇区，每区分4小块。1块有16字节内容。编程时一写就要写16字节或者一读就读入16字节。

|  |  |  |
| --- | --- | --- |
| 扇区0 | 块0 | 放序列号和厂商号，固定没得改 |
| 块1 | 内容 |
| 块2 | 内容 |
| 块3 | 扇区设置码 |
| 扇区1 | 块0 | 内容 |
| 块1 | 内容 |
| 块2 | 内容 |
| 块3 | 扇区设置码 |
| ... | ... | ... |
| 扇区15 | 块0 | 内容 |
| 块1 | 内容 |
| 块2 | 内容 |
| 块3 | 扇区设置码 |

除了块3是用于扇区设置不能随意写，其他的块内容就是随意数据读写。

读写块时需要用扇区密码认证，密码正确才可以读写。扇区密码有A、B两种，都可以用。

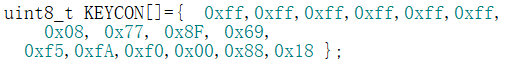
一张新卡默认的A、B两密码都是{0xff, 0xff, 0xff, 0xff, 0xff, 0xff}。

块3是扇区设置码，设置了这个扇区的密码A、B，设置了这个扇区的每块的读写权限。扇区设置码16字节组成如下：



左边的是扇区A密码，右边是扇区B密码。中间的则是权限设置。

中间的权限设置怎样写这4字节，他的加密算法很繁杂。干脆直接在块3写入：



这样设置后，该扇区的密码A保持默认，密码B改为{0xf5, 0xfa, 0xf0, 0x00, 0x88, 0x18}。

中间的08 77 8F 69，表示该扇区使用A密码可以读块0块1块2块3；若使用B密码则可读可写块0块1块2块3，但块3的16字节中读到的扇区AB密码内容都只会显示0，只能读到中间4字节是多少。

编程中用到的块地址取值0~63。是从扇区0块0的0，往下直数到扇区15块3的63。

#### 4）编程：

程序文件夹为“写卡”，使用KEIL5，stm32f4的裸机程序，可烧录到空闲的读卡器板或F411最小系统板来使用。

文件rcc522\_config用来配置SPI，先在头文件配置好使用的SPI，读卡器的板使用SPI2，F411小板则用SPI1。文件rcc522\_function里的是与RCC522读卡有关的所有函数。

读卡的步骤如下：

首先使用char PcdRequest ( uint8\_t ucReq\_code, uint8\_t \* pTagType ) 函数，返回值是MI\_OK的话就是检测到有卡。pTagType的地址取值会写入卡的种类信息。

当上面的返回值是MI\_OK，检测到有卡，接着使用char PcdAnticoll ( uint8\_t \* pSnr )函数，防冲撞，避免不小心多张卡一起靠近读卡器的情况产生问题。返回值是MI\_OK则可以进入下一步。pSnr的取值会写入这张卡读取到的卡序列号。

下一步是char PcdAuthState ( uint8\_t ucAuth\_mode, uint8\_t ucAddr, uint8\_t \* pKey, uint8\_t \* pSnr )函数。验证卡的密码。ucAuth\_mode选择认证A密码还是B密码，ucAddr是想要读写的块地址。pKey是密码。pSnr是上面防冲撞时拿到的卡序列号。验证成功的话会返回MI\_OK，这时候就可以对这个块进行读写了。

写块：char PcdWrite ( uint8\_t ucAddr, uint8\_t \* pData )

读块：char PcdRead ( uint8\_t ucAddr, uint8\_t \* pData )

#### 5）使用操作

在rc522\_config.h修改好SPI配置，是125行到171行，想用哪个就取消注释哪个，然后把另一个注释掉。

在main.c的120行修改val的值，为想写入卡的枪IP号。

第一步注释掉step 2的内容，取消注释step 1，然后烧录写卡一次。然后注释step 2，取消注释step 1，再烧录写卡一次即可。

烧录好程序进读卡器板或F411小黑板，小黑板和RC522模块连接好线。建议进入KEIL5的DEBUG监视仿真模式，运行监视看apple变量的值变化。apple值变化就是成功写卡了。

### 无线模块

无线模块需要先初始化，把波特率改为115200，修改发射功率最大，并把信道和PID都初始化作0。程序在“初始化无线模块”文件夹，使用的是KEIL5，裸机程序。

新模块初始化步骤，首先在bsp\_usartx.h文件中，把第12行USARTx\_BAUDRATE修改作9600。连接烧录器和读卡器板，开启KEIL5的DEBUG模式，运行。在监视窗口便可看到情况了。若成功改好，apple变量由0变为1，aRxBuffer缓存中放入接收到的模块回复的几个OK信息。

初始化成功后，此程序也可以用作测试模块或对模块进行其他操作。首先修改bsp\_usartx.h文件第12行，改为115200。然后注释掉main.c文件中的122行到141行，取消注释108行到119行，在里面进行想要的测试操作。照样可在DEBUG时的aRxBuff看模块回复的消息。

### 3.3 读卡器

#### 组成：

读卡器程序由LED灯、按键、RC522读卡、RS232串口连无线模块这四部分内容组成。使用STM32F401RCT6，带RT-Thread操作系统的程序。

RT-Thread项目在文件夹“读卡器主项目”中，解压后用RT-Thread Studio导入项目。

也可从git@github.com:hafuli123/AcGunCardReader.git下载读卡器的RT-Thread项目程序。

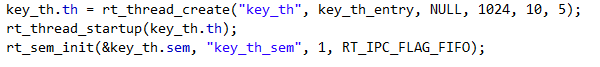
#### 按键程序和特殊GPIO程序：

看STM32单片机的电路图图1，从左看，KEY\_ON是右边开关机按键，KEY2是左边按键。

开关机：电路图右边的PWR\_EN，程序写这个GPIO输出高低电平控制开关机。先什么都不干，按下KEY\_ON按钮，开机了，但手一松开这个按钮就马上关机了。这时若在main程序开头给PWR\_EN写入高电平，这样手松开KEY\_ON按键，就不会马上自己熄灭关机了。关机时，检测KEY\_ON按下了一段时间，把PWR\_EN置为0，则松开KEY\_ON按钮他就会关机了。

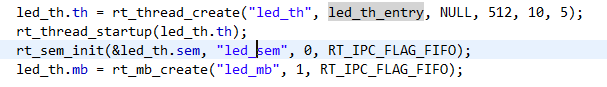
特殊的GPIO：单片机电路图左下边的BAT\_DONE和BAT\_CHRG用来检测充电。当读卡器typec充电口插入线充电，BAT\_CHRG会读取到高电平。若电池没有充满电，BAT\_DONE会读到低电平。而充满电了BAT\_DONE会读到高电平。

在RTThread项目中，main.c文件里，由子线程key\_th进行编程实现。



#### LED灯程序

读卡器有3个LED灯，绿色黄色红色。子线程led\_th控制哪个灯亮，把想亮的灯写入邮件发送过去，子线程就可根据邮件内容来控制灯。



比如想亮绿灯，可如下：



项目中把灯的邮件内容这样记作，led\_x为长亮，led\_xsk为先闪一下再长亮：



读卡器的LED灯操作是如下：当正常状态，没有卡靠近，此时会亮绿灯。卡靠近会亮黄灯，则表明此卡被成功识别并开始进行读卡器修改靶面板地址的操作，完成后变回绿灯。若发现卡靠近后变黄，一会后先红一下再变绿，这是修改失败了。按下左按钮，绿灯变黄灯，此时读卡器的状态改变，这时可以从靶面发送地址信息修改读卡器的地址。插入TYPEC线时绿灯变红灯，若充电完成了则变回绿灯。

#### RC522读卡

RC522读卡需要以下子线程：

首先rc522\_th子线程用于读卡，当有卡检测到并成功读取时，触发gc\_send\_th子线程。

gc\_send\_th子线程用来判别读到的卡的枪地址信息，若本身就和读卡器的地址一样，即是和靶面的地址一样，就不用改什么了。若是不同，则发送3字节的枪卡信息数组给靶面，读卡器自己也开始修改自己的地址。

绝对注意信息间的差别：读卡器读到的信息一定是16字节的数组，而读卡器和靶面传输的信息则是3字节的数组。

绝对要注意IP地址，比如IP是0x1234，则无线模块的信道为0x12，PID为0x34，但注意数组中是发0XAA , 0X34 , 0X12 的顺序。

#### RS232串口连无线模块

这是读卡器最繁杂复杂的程序内容。

需要用到两个子线程和一个变量：

gc\_recv\_th子线程一直循环，每隔1ms拿一次数据看USART有没有信息，没有就算了，有就存入rxbuf缓存。

通信中使用gc24\_stat这个变量发邮件来控制步骤。

当单片机USART接收到无线模块传来的信息，gc\_recv\_th子线程把信息存入rxbuf，通过对rxbuf里信息的判别，来控制gc24\_stat的值，然后发送邮件把gc24\_stat传给gc\_ioctl\_th子线程。

gc\_ioctl\_th子线程根据收到的gc24\_stat，来决定下一步单片机要向无线模块发送什么信息。

无线模块需要的工作有三部分，第一部分为获取初始读卡器的IP信息；第二部分为读卡修改靶面IP时的通信；第三部分是靶面向读卡器修改IP地址时的通信。

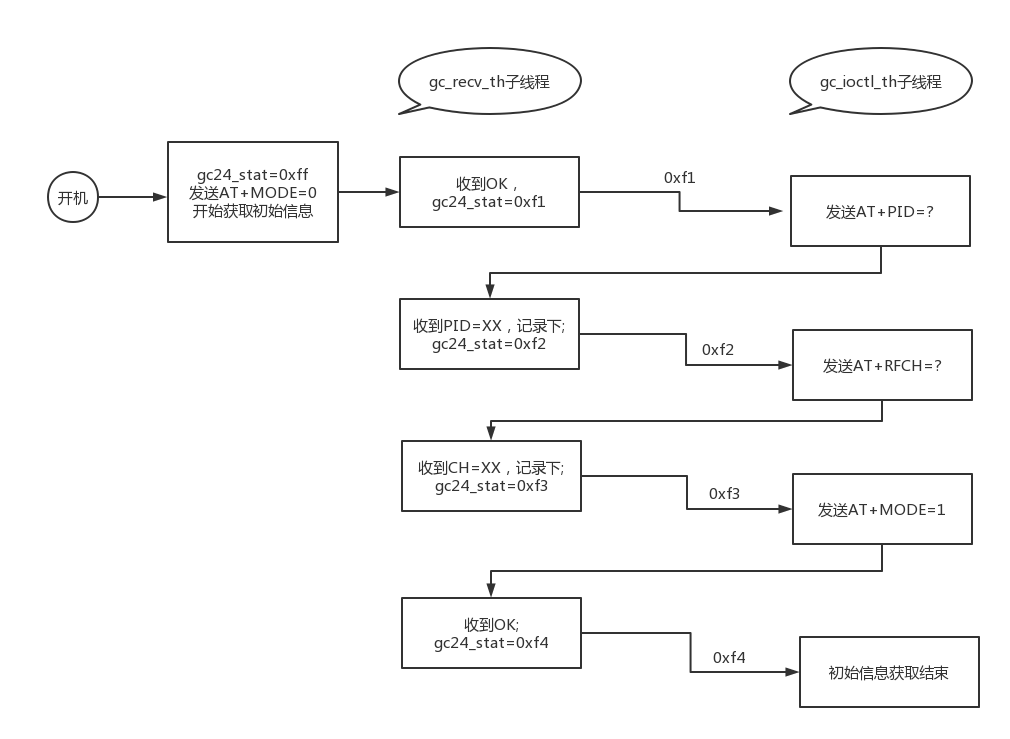
第一部分获取读卡器初始信息。开机后，读卡器需要知道自己无线模块的信道和PID信息，并记录入IP。因此需要通过串口给无线模块发送以下命令：

AT+MODE=0\r\n -- 无线模块从透传模式进入AT指令模式；

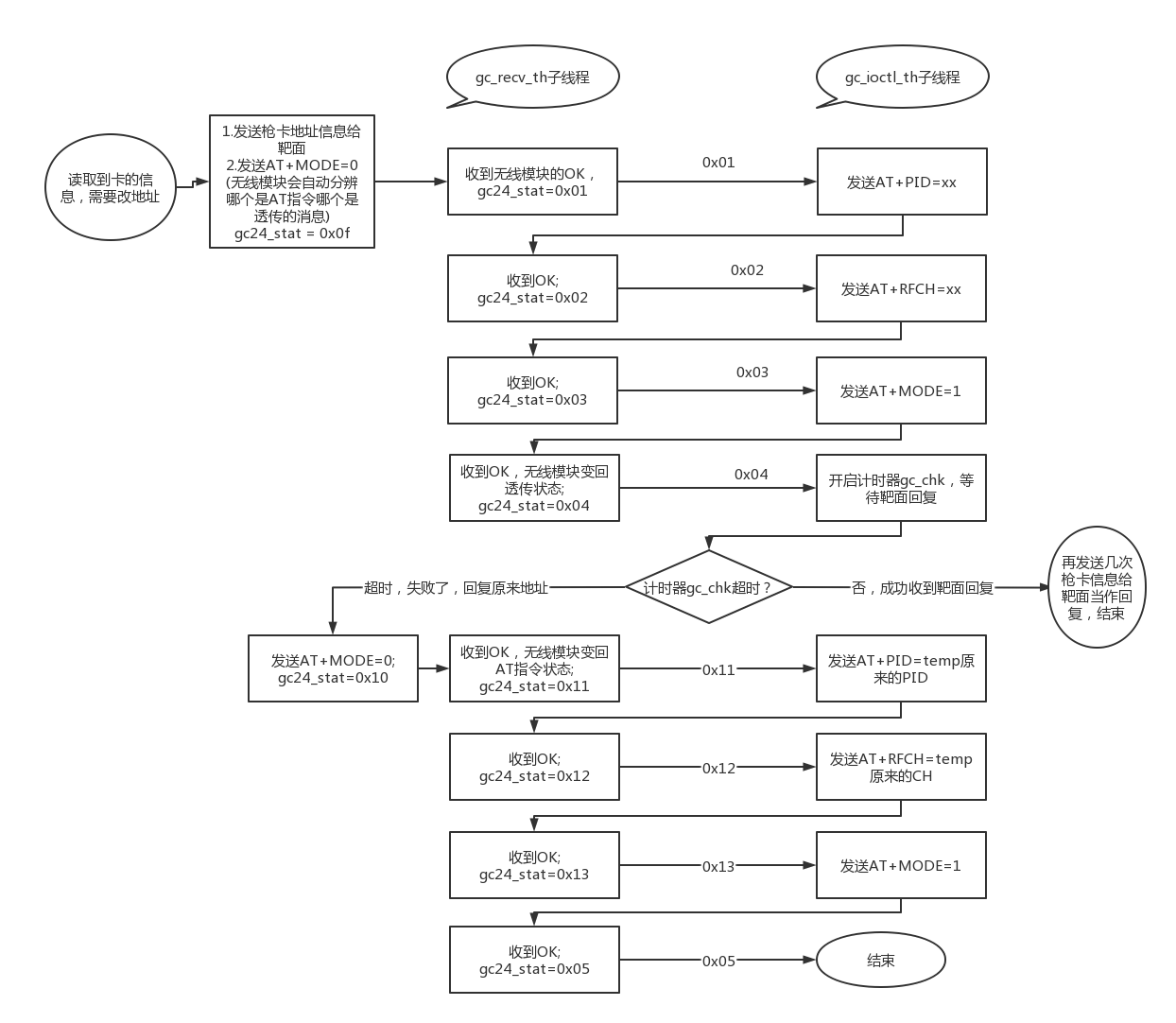
AT+PID=?\r\n -- 询问无线模块的PID是多少；

AT+RFCH=?\r\n -- 询问无线模块的信道是多少；

AT+MODE=1\r\n -- 无线模块进入透传模块，此时可以与其他模块透传信息；



第二部分是读到卡后与靶面的通信。具体的步骤是读卡器发送枪卡信息给靶面，一发送后自己马上修改地址改成与枪卡一样的地址。这时候开始等待靶面的回复。靶面那边改好地址了，会回复信息给读卡器。读卡器收到这个消息，表示读卡器和靶面两者都改好了能互相用新地址通信了，回复几次信息给靶就结束。若是没收到靶面的回复，表示失败了，读卡器会回退回复原来的地址。（图可用WPS“图片预览”看清晰大图）



第三部分是靶面向读卡器发送地址信息。用户可以通过IPAD把要改的地址信息发给靶面，靶面再把信息发给读卡器。

