RFID-RC522 读卡模块使用教程

1.1 RFID-RC522 模块简介

MFRC522 是高度集成的非接触式（13.56MHz）读写卡芯片。此发送模块利用调制和解调的原理，并将它们完全集成到各种非接触式通信方法和协议中（13.56MHz）。MFRC522 发送模块支持下面的工作模式：

读写器，支持 ISO14443A/MIFARE

MFRC522 的内部发送器部分可驱动读写器天线与 ISO14443A/MIFARE 卡和应答机的通信，无需其它的电路。接收器部分提供一个功能强大和高效的解调和译码电路，用来处理兼容 ISO14443A/MIFARE 的卡和应答机的信号。数字电路部分处理完整的 ISO14443A 帧和错误检测（奇偶&CRC）。MFRC522 支持 MIFARE Classic（如，MIFARE 标准）器件。MFRC522 支持 MIFARE 更高速的非接触式通信，双向数据传输速率高达 424kbit/s。

可实现各种不同主机接口的功能：

SPI 接口

串行 UART（类似 RS232，电压电平取决于提供的管脚电压）

I2C 接口

该模块的外形和引脚图如图 1.1.1 所示：

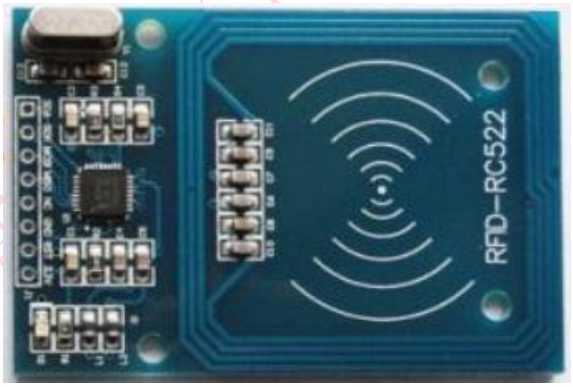


图 1.1.1 RFID-RC522 模块外观引脚图

模块引出了 8 个管脚供用户使用，对应的管脚功能如下：

3.3V：电源输入端，电压为 3.3V。

RST：模块复位管脚

GND：GND 端

NC（IRQ）：空管脚

MISO：SPI 接口主入从出管脚

MOSI：SPI 接口主出从入管脚

SCK：时钟管脚

SDA：数据管脚

模块详细资料可以参考“\RFID-RC522 门禁控制系统\参考资料”内文档。

1.2 硬件设计

开发板上并没有预留 RFID-RC522 模块接口,但可以使用杜邦线连接此模块，RFID-RC522 模块与开发板 STM32 芯片管脚连接说明如下：

管脚接线图：

3.3V---3.3V

RST---PF4

GND---GND

NC(IRQ)---悬空

MISO---PF3

MOSI---PF2

SCK---PF1

NSS(SDA)--PF0

1.3 软件设计

打开“\RFID-RC522 门禁控制系统\RFID-RC522 门禁控制系统程序”工程，可以看到我们加入了 RC522.c 源文件和 RC522.h 头文件，所有 RFID-RC522 相关的驱动代码和定义都在这两个文件中实现。同时为了实现开锁后自动关锁，我们还加入了 STM32 定时器驱动文件 tim.c 和 tim.h 头文件,软件设计中为了考虑能够移植到其他单片机中，采用单片机管脚模拟 SPI 时序的方式进行编程。

1.3.1 RC522 驱动程序

要想对模块内部的数据块进行读写，需要完成 4 个步骤：寻卡→防冲突→选卡→读/写卡，这些在 RC522.c 文件内都有注释，由于驱动代码比较多，这里我们只复制部分重要代码，详细代码大家可以打开工程查看，主要代码如下：

```
#include "RC522.h"

#define MAXRLEN 18

void RC522_Init(void)
{
   GPIO_InitTypeDef  GPIO_InitStructure;
   RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOF, ENABLE);        //
使能 PF 端口时钟

   GPIO_InitStructure.GPIO_Pin                                  =
GPIO_Pin_1|GPIO_Pin_0|GPIO_Pin_2|GPIO_Pin_4;
   GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;       //推挽输
出
   GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;      //IO 口速
```

```
度为 50MHz
       GPIO_Init(GPIOF, &GPIO_InitStructure);
       GPIO_SetBits(GPIOF,GPIO_Pin_1|GPIO_Pin_0|GPIO_Pin_2|GPIO_Pin_4)
;  //拉高

       GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
       GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;      //上拉输入

       GPIO_Init(GPIOF, &GPIO_InitStructure);
    }



    ////////////////////////////////////////////////////////////////
////
    //功    能：寻卡
    //参数说明: req_code[IN]:寻卡方式
    //                0x52 = 寻感应区内所有符合 14443A 标准的卡
    //                0x26 = 寻未进入休眠状态的卡
    //          pTagType[OUT]：卡片类型代码
    //                0x4400 = Mifare_UltraLight
    //                0x0400 = Mifare_One(S50)
    //                0x0200 = Mifare_One(S70)
    //                0x0800 = Mifare_Pro(X)
    //                0x4403 = Mifare_DESFire
    //返    回: 成功返回 MI_OK
    ////////////////////////////////////////////////////////////////
////
    char PcdRequest(unsigned char req_code,unsigned char *pTagType)
    {
       char status;
       unsigned int  unLen;
       unsigned char ucComMF522Buf[MAXRLEN];

       ClearBitMask(Status2Reg,0x08);
       WriteRawRC(BitFramingReg,0x07);
       SetBitMask(TxControlReg,0x03);

       ucComMF522Buf[0] = req_code;

       status                                                    =
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,1,ucComMF522Buf,&unLen);

       if ((status == MI_OK) && (unLen == 0x10))
```

```
        {
            *pTagType     = ucComMF522Buf[0];
            *(pTagType+1) = ucComMF522Buf[1];
        }
        else
        {   status = MI_ERR;
        }

        return status;
    }

    ////////////////////////////////////////////////////////////////////
////
    //功    能：防冲撞
    //参数说明: pSnr[OUT]:卡片序列号，4 字节
    //返    回: 成功返回 MI_OK
    ////////////////////////////////////////////////////////////////////
////
    char PcdAnticoll(unsigned char *pSnr)
    {
        char status;
        unsigned char i,snr_check=0;
        unsigned int  unLen;
        unsigned char ucComMF522Buf[MAXRLEN];


        ClearBitMask(Status2Reg,0x08);
        WriteRawRC(BitFramingReg,0x00);
        ClearBitMask(CollReg,0x80);

        ucComMF522Buf[0] = PICC_ANTICOLL1;
        ucComMF522Buf[1] = 0x20;

        status                                                          =
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,2,ucComMF522Buf,&unLen);

        if (status == MI_OK)
        {
            for (i=0; i<4; i++)
              {
                  *(pSnr+i)  = ucComMF522Buf[i];
                  snr_check ^= ucComMF522Buf[i];

              }
```

```
            if (snr_check != ucComMF522Buf[i])
            {   status = MI_ERR;    }
        }

        SetBitMask(CollReg,0x80);
        return status;
    }

    /////////////////////////////////////////////////////////////////////
////
    //功    能：选定卡片
    //参数说明: pSnr[IN]:卡片序列号，4 字节
    //返    回: 成功返回 MI_OK
    /////////////////////////////////////////////////////////////////////
////
    char PcdSelect(unsigned char *pSnr)
    {
        char status;
        unsigned char i;
        unsigned int  unLen;
        unsigned char ucComMF522Buf[MAXRLEN];

        ucComMF522Buf[0] = PICC_ANTICOLL1;
        ucComMF522Buf[1] = 0x70;
        ucComMF522Buf[6] = 0;
        for (i=0; i<4; i++)
        {
          ucComMF522Buf[i+2] = *(pSnr+i);
          ucComMF522Buf[6]  ^= *(pSnr+i);
        }
        CalulateCRC(ucComMF522Buf,7,&ucComMF522Buf[7]);

        ClearBitMask(Status2Reg,0x08);

        status                                                          =
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,9,ucComMF522Buf,&unLen);

        if ((status == MI_OK) && (unLen == 0x18))
        {   status = MI_OK;  }
        else
        {   status = MI_ERR;    }

        return status;
    }
```

```
    /////////////////////////////////////////////////////////////////////
////
    //功    能：验证卡片密码
    //参数说明: auth_mode[IN]: 密码验证模式
    //                 0x60 = 验证 A 密钥
    //                 0x61 = 验证 B 密钥
    //          addr[IN]：块地址
    //          pKey[IN]：密码
    //          pSnr[IN]：卡片序列号，4 字节
    //返    回: 成功返回 MI_OK
    /////////////////////////////////////////////////////////////////////
////
    char    PcdAuthState(unsigned    char    auth_mode,unsigned    char
addr,unsigned char *pKey,unsigned char *pSnr)
    {
        char status;
        unsigned int  unLen;
        unsigned char i,ucComMF522Buf[MAXRLEN];

        ucComMF522Buf[0] = auth_mode;
        ucComMF522Buf[1] = addr;
        for (i=0; i<6; i++)
        {    ucComMF522Buf[i+2] = *(pKey+i);   }
        for (i=0; i<6; i++)
        {    ucComMF522Buf[i+8] = *(pSnr+i);   }
     //   memcpy(&ucComMF522Buf[2], pKey, 6);
     //   memcpy(&ucComMF522Buf[8], pSnr, 4);

        status                                                          =
PcdComMF522(PCD_AUTHENT,ucComMF522Buf,12,ucComMF522Buf,&unLen);
        if ((status != MI_OK) || (!(ReadRawRC(Status2Reg) & 0x08)))
        {   status = MI_ERR;   }

        return status;
    }

    /////////////////////////////////////////////////////////////////////
////
    //功    能：读取 M1 卡一块数据
    //参数说明: addr[IN]：块地址
    //          pData[OUT]：读出的数据，16 字节
    //返    回: 成功返回 MI_OK
    /////////////////////////////////////////////////////////////////////
```

```
////
    char PcdRead(unsigned char addr,unsigned char *pData)
    {
        char status;
        unsigned int  unLen;
        unsigned char i,ucComMF522Buf[MAXRLEN];

        ucComMF522Buf[0] = PICC_READ;
        ucComMF522Buf[1] = addr;
        CalulateCRC(ucComMF522Buf,2,&ucComMF522Buf[2]);

        status                                                    =
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,4,ucComMF522Buf,&unLen);
        if ((status == MI_OK) && (unLen == 0x90))
     //   {   memcpy(pData, ucComMF522Buf, 16);   }
        {
            for (i=0; i<16; i++)
            {    *(pData+i) = ucComMF522Buf[i];   }
        }
        else
        {   status = MI_ERR;   }

        return status;
    }

    ////////////////////////////////////////////////////////////////
////
    //功    能：写数据到 M1 卡一块
    //参数说明: addr[IN]：块地址
    //          pData[IN]：写入的数据，16 字节
    //返    回: 成功返回 MI_OK
    ////////////////////////////////////////////////////////////////
////
    char PcdWrite(unsigned char addr,unsigned char *pData)
    {
        char status;
        unsigned int  unLen;
        unsigned char i,ucComMF522Buf[MAXRLEN];

        ucComMF522Buf[0] = PICC_WRITE;
        ucComMF522Buf[1] = addr;
        CalulateCRC(ucComMF522Buf,2,&ucComMF522Buf[2]);

        status                                                    =
```

```
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,4,ucComMF522Buf,&unLen);

        if ((status != MI_OK) || (unLen != 4) || ((ucComMF522Buf[0] &
0x0F) != 0x0A))
        {   status = MI_ERR;   }

        if (status == MI_OK)
        {
            //memcpy(ucComMF522Buf, pData, 16);
            for (i=0; i<16; i++)
            {    ucComMF522Buf[i] = *(pData+i);   }
            CalulateCRC(ucComMF522Buf,16,&ucComMF522Buf[16]);

            status                                                  =
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,18,ucComMF522Buf,&unLen);
            if ((status != MI_OK) || (unLen != 4) || ((ucComMF522Buf[0]
& 0x0F) != 0x0A))
            {   status = MI_ERR;   }
        }

        return status;
    }
```

接下来我们看看 RC522.h 代码，该头文件主要定义了一些 RC522 的命令字，模块管脚定义以及函数声明，如果需要修改对应的 IO 口，可以把此部分管脚定义和 RC522.c 管脚端口初始化修改，代码如下：

```
    #define MF522_NSS PFout(0)  //PF0   SDA
    #define MF522_SCK PFout(1)  //PF1
    #define MF522_SI  PFout(2)  //PF2
    #define MF522_SO  PFin(3)   //PF3
    #define MF522_RST PFout(4)  //PF4

    void RC522_Init(void)
    {
       GPIO_InitTypeDef  GPIO_InitStructure;
       RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOF, ENABLE);       //
使能 PF 端口时钟

       GPIO_InitStructure.GPIO_Pin                                 =
GPIO_Pin_1|GPIO_Pin_0|GPIO_Pin_2|GPIO_Pin_4;
       GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;      //推挽输
出
       GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;     //IO 口速
度为 50MHz
       GPIO_Init(GPIOF, &GPIO_InitStructure);
```

```
    GPIO_SetBits(GPIOF,GPIO_Pin_1|GPIO_Pin_0|GPIO_Pin_2|GPIO_Pin_4)
;  //拉高

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;      //上拉输入

    GPIO_Init(GPIOF, &GPIO_InitStructure);
}
```

定时器初始化及中断函数在我们库函数实验例程内已经介绍过，这里就不重复。

1.3.2 主函数

打开 main.c，代码如下：

```
/* 本程序使用的是 RFID-RC522 射频模块设计的一个门禁系统，当感应卡放到射频模块区域内会感应到
    卡，如果卡序列号和程序设计一致就会认为是正确开锁，D2 指示灯亮，LCD 上显示开锁，5 秒钟以后
    自动关锁，D2 指示灯灭。当卡错误时候不会显示，D2 也不会亮。卡的序列号是唯一的。

    管脚接线图：
    RST---PF4
    MISO---PF3
    MOSI---PF2
    SCK---PF1
    NSS(SDA)--PF0

    */

#include "system.h"
#include "SysTick.h"
#include "led.h"
#include "usart.h"
#include "tftlcd.h"
#include "RC522.h"
#include "time.h"


unsigned char data1[16] = {0x12,0x34,0x56,0x78,0xED,0xCB,0xA9,0x87,0x12,0x34,0x56,0x78,0x01,0xFE,0x01,0xFE};
//M1 卡的某一块写为如下格式，则该块为钱包，可接收扣款和充值命令
//4 字节金额（低字节在前）+4 字节金额取反+4 字节金额+1 字节块地址
```

＋1 字节块地址取反＋1 字节块地址＋1 字节块地址取反

```
    unsigned char data2[4]  = {0,0,0,0x01};
    unsigned char DefaultKey[6] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};


    unsigned char g_ucTempbuf[20];

    int main()
    {
       unsigned char status,i;
       unsigned int temp;

       SysTick_Init(72);
       NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);  //中断优先级
分组 分 2 组
       LED_Init();
       USART1_Init(9600);
       TFTLCD_Init();           //LCD 初始化


       FRONT_COLOR=GREEN;
       LCD_ShowString(10,10,tftlcd_data.width,tftlcd_data.height,16,"
PRECHIN STM32F1");
       LCD_ShowString(10,30,tftlcd_data.width,tftlcd_data.height,16,"
www.prechin.net");
       LCD_ShowString(10,50,tftlcd_data.width,tftlcd_data.height,16,"
RFID-RC522 Test");
       FRONT_COLOR=RED;
       LCD_ShowString(10,110,tftlcd_data.width,tftlcd_data.height,16,
"Close Door...");

       RC522_Init();
       PcdReset();
       PcdAntennaOff();
       delay_ms(10);
       PcdAntennaOn();
       delay_ms(10);

       TIM4_Init(1000,7199);
       printf("Start \r\n");

       while ( 1 )
         {
            status = PcdRequest(PICC_REQALL, g_ucTempbuf);//寻卡
```

```
            if (status != MI_OK)
            {
              PcdReset();
              PcdAntennaOff();
              PcdAntennaOn();
             continue;
            }
          printf("卡的类型:");
          for(i=0;i<2;i++)
          {
              temp=g_ucTempbuf[i];
              printf("%X",temp);

          }
            status = PcdAnticoll(g_ucTempbuf);//防冲撞
            if(status != MI_OK)
            {    continue;    }

          ////////以下为超级终端打印出的内容//////////////////////

          printf("卡序列号：");    //超级终端显示,
          for(i=0;i<4;i++)
          {
              temp=g_ucTempbuf[i];
              printf("%X",temp);

          }


    if(g_ucTempbuf[0]==0xd4&&g_ucTempbuf[1]==0xd5&&g_ucTempbuf[2]==0x
34&&g_ucTempbuf[3]==0x00)
          {
              led2=0;
              FRONT_COLOR=RED;

    LCD_ShowString(10,110,tftlcd_data.width,tftlcd_data.height,16,"Op
en Door...  ");//开门
          }
          else
          {
              led2=1;
              FRONT_COLOR=RED;

    LCD_ShowString(10,110,tftlcd_data.width,tftlcd_data.height,16,"
```

```
");
            }


    ////////////////////////////////////////////////////////////

              status = PcdSelect(g_ucTempbuf);//选定卡片
              if (status != MI_OK)
              {    continue;    }

                 status  =  PcdAuthState(PICC_AUTHENT1A,  1,  DefaultKey,
g_ucTempbuf);//验证卡片密码
              if (status != MI_OK)
              {    continue;    }

              status = PcdWrite(1, data1);//写块
               if (status != MI_OK)
               {    continue;    }

            while(1)
            {
                status = PcdRequest(PICC_REQALL, g_ucTempbuf);//寻卡
                if (status != MI_OK)
                {

                     PcdReset();
                     PcdAntennaOff();
                     PcdAntennaOn();
                   continue;
                }
                 status = PcdAnticoll(g_ucTempbuf);//防冲撞
                  if (status != MI_OK)
                  {    continue;    }

                status = PcdSelect(g_ucTempbuf);//选定卡片
                  if (status != MI_OK)
                  {    continue;    }

                   status = PcdAuthState(PICC_AUTHENT1A, 1, DefaultKey,
g_ucTempbuf);//验证卡片密码
                  if (status != MI_OK)
                  {    continue;    }
```

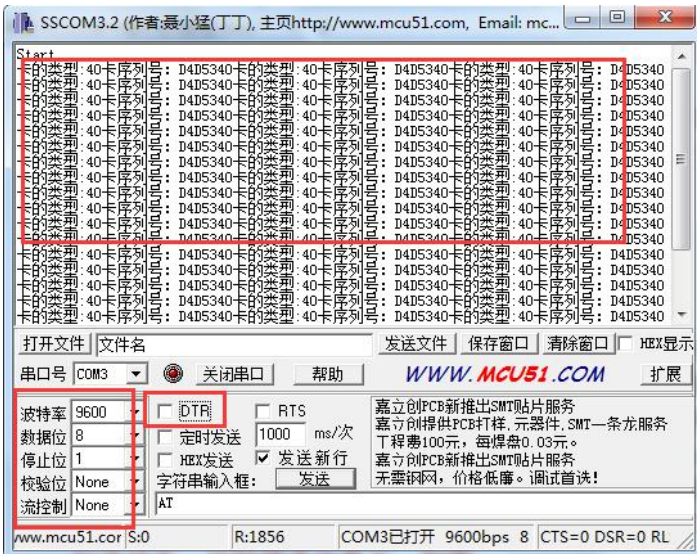
```
                status = PcdValue(PICC_DECREMENT,1,data2);//扣款
                if (status != MI_OK)
                {    continue;    }

                status = PcdBakValue(1, 2);//块备份
                if (status != MI_OK)
                {    continue;    }

                status = PcdRead(2, g_ucTempbuf);//读块
                if (status != MI_OK)
                {    continue;    }
            printf("卡读块：");  //超级终端显示,
            for(i=0;i<16;i++)
            {
                temp=g_ucTempbuf[i];
                printf("%X",temp);

            }

            printf("\n");
            PcdHalt();
        }
    }
}
```

程序运行时先把使用到的硬件端口及时钟初始化，为了调试方便，我们还初始化了串口，可以打开串口调试助手查看 RFID-RC522 的输出信息，注意串口助手波特率设置为 9600，打开串口助手后先勾选 DTR 复选框，然后再取消 DTR 勾选状态，即可输出信息到串口助手上显示。

按照寻卡→防冲突→选卡→读/写卡操作步骤读取磁卡序列号，比较序列号判断磁卡是否正确，同时将磁卡的类型及序列号通过串口发送到串口助手上显示。

1.4 实验现象

将模块连接好以后，把实验程序下载到开发板内即可看到 LCD 显示门禁状态，D1 指示灯不断闪烁，表示程序运行。当把磁卡触碰下 RFID-RC522 模块感应区时，LCD 显示“Open Door...”，同时指示灯 D2 点亮，表示门禁系统开锁，5 秒钟后门禁系统关闭，D2 指示灯灭，LCD 上显示“Close Door...”。如果打开串口调试助手可以看到输出磁卡的类型及序列号，但是当打开串口调试助手时一定要将波特率设置为 9600，打开串口助手后先勾选 DTR 复选框，然后再取消 DT 勾选状态，这时候才会显示。如图：



如果你连接好模块，发现用磁卡触碰下感应区 LCD 上并没有显示“Open Door...”，那可能是你的磁卡序列号与我们程序中设置的不一样，这个时候可以通过串口调试助手上输出的序列号数据来修改程序中的序列号，需修改序列号代码在主函数中，如下：

```
92          if(g_ucTempbuf[0]==0xd4&&g_ucTempbuf[1]==0xd5&&g_ucTempbuf[2]==0x34&&g_ucTempbuf[3
93          {
94              led2=0;
95              FRONT_COLOR=RED;
96              LCD_ShowString(10,110,tftlcd_data.width,tftlcd_data.height,16,"Open Door...  "
97          }
98          else
99          {
100             led2=1;
101             FRONT_COLOR=RED;
102             LCD_ShowString(10,110,tftlcd_data.width,tftlcd_data.height,16,"
103         }
104
```

从代码中可以看到，我们判断语句中使用的序列号正好是串口中输出的序列号，所以你可以把你磁卡输出的序列号替换下即可。