RFID读卡器实验

1）概述

射频技术也简称 RFID,RFID 是英文 radio frequency identification”的缩写，叫做射频识别技术，简称射频技术。

RFID 工作原理

射频识别系统的基本模型如图所示。

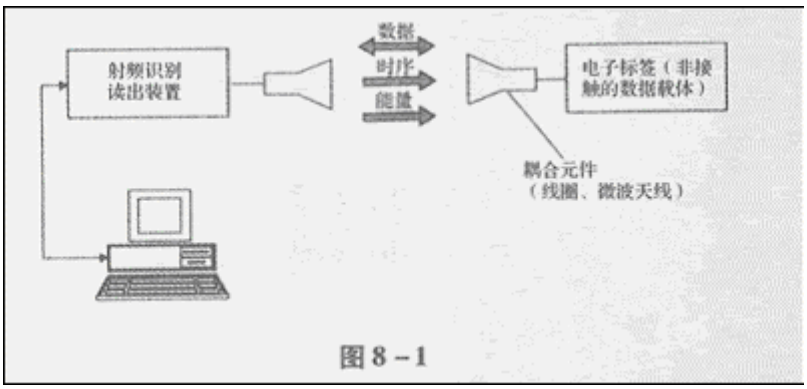
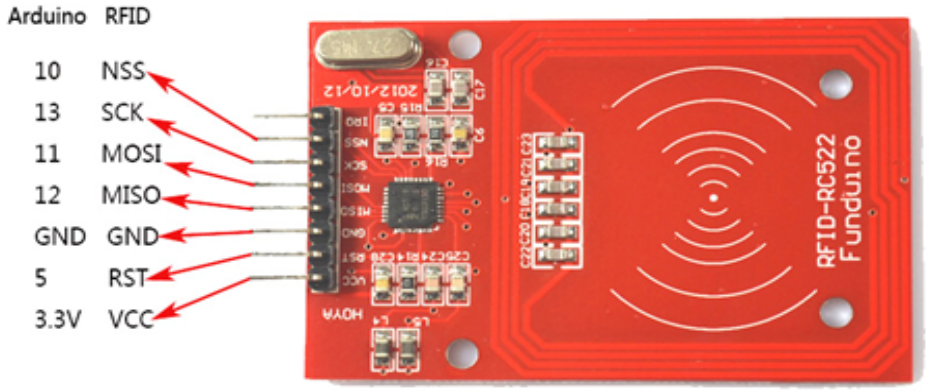


图 8－1

其中，电子标签又称为射频标签、应答器、数据载体；阅读器又称为读出装置，扫描器、通讯器、读写器(取决于电子标签是否可以无线改写数据)。电子标签与阅读器之间通过耦合元件实现射频信号的空间(无接触)耦合、在耦合通道内，根据时序关系，实现能量的传递、数据的交换。

本模块，大家一定要使用+3.3V 供电，否则会烧掉模块 。

2）具体的连线图如下。



4) 程序代码：

```
#include <SPI.h>

#define    uchar      unsigned char
#define    uint  unsigned int

//数组最大长度
#define MAX_LEN 16

//////////////////////////////////////////////////////////////////////
//set the pin
//////////////////////////////////////////////////////////////////////
const int chipSelectPin = 10; //如果控制板为 UNO,328,168
const int chipSelectPin = 53; //如果控制板为 mega 2560,1280
const int NRSTPD = 5;

//MF522 命令字
#define PCD_IDLE              0x00               //NO action;取消当前命令
#define PCD_AUTHENT           0x0E               //验证密钥
#define PCD_RECEIVE           0x08               //接收数据
#define PCD_TRANSMIT          0x04               //发送数据
#define PCD_TRANSCEIVE        0x0C               //发送并接收数据
#define PCD_RESETPHASE        0x0F               //复位
#define PCD_CALCCRC           0x03               //CRC 计算

//Mifare_One 卡片命令字
#define PICC_REQIDL           0x26               //寻天线区内未进入休眠状态
#define PICC_REQALL           0x52               //寻天线区内全部卡
#define PICC_ANTICOLL         0x93               //防冲撞
#define PICC_SElECTTAG        0x93               //选卡
#define PICC_AUTHENT1A        0x60               //验证 A 密钥
#define PICC_AUTHENT1B        0x61               //验证 B 密钥
#define PICC_READ             0x30               //读块
#define PICC_WRITE            0xA0               //写块
#define PICC_DECREMENT        0xC0               //
#define PICC_INCREMENT        0xC1               //
#define PICC_RESTORE          0xC2               //调块数据到缓冲区
#define PICC_TRANSFER         0xB0               //保存缓冲区中数据
#define PICC_HALT             0x50               //休眠


//和 MF522 通讯时返回的错误代码
#define MI_OK                 0
#define MI_NOTAGERR           1
#define MI_ERR                2
```

```
//------------------MFRC522 寄存器----------------
//Page 0:Command and Status
#define     Reserved00              0x00
#define     CommandReg              0x01
#define     CommIEnReg              0x02
#define     DivlEnReg               0x03
#define     CommIrqReg              0x04
#define     DivIrqReg               0x05
#define     ErrorReg                0x06
#define     Status1Reg              0x07
#define     Status2Reg              0x08
#define     FIFODataReg             0x09
#define     FIFOLevelReg            0x0A
#define     WaterLevelReg           0x0B
#define     ControlReg              0x0C
#define     BitFramingReg           0x0D
#define     CollReg                 0x0E
#define     Reserved01              0x0F
//Page 1:Command
#define     Reserved10              0x10
#define     ModeReg                 0x11
#define     TxModeReg               0x12
#define     RxModeReg               0x13
#define     TxControlReg            0x14
#define     TxAutoReg               0x15
#define     TxSelReg                0x16
#define     RxSelReg                0x17
#define     RxThresholdReg          0x18
#define     DemodReg                0x19
#define     Reserved11              0x1A
#define     Reserved12              0x1B
#define     MifareReg               0x1C
#define     Reserved13              0x1D
#define     Reserved14              0x1E
#define     SerialSpeedReg          0x1F
//Page 2:CFG
#define     Reserved20              0x20
#define     CRCResultRegM           0x21
#define     CRCResultRegL           0x22
#define     Reserved21              0x23
#define     ModWidthReg             0x24
#define     Reserved22              0x25
```

```
#define     RFCfgReg                0x26
#define     GsNReg                  0x27
#define     CWGsPReg                0x28
#define     ModGsPReg               0x29
#define     TModeReg                0x2A
#define     TPrescalerReg           0x2B
#define     TReloadRegH             0x2C
#define     TReloadRegL             0x2D
#define     TCounterValueRegH       0x2E
#define     TCounterValueRegL       0x2F
//Page 3:TestRegister
#define     Reserved30              0x30
#define     TestSel1Reg             0x31
#define     TestSel2Reg             0x32
#define     TestPinEnReg            0x33
#define     TestPinValueReg         0x34
#define     TestBusReg              0x35
#define     AutoTestReg             0x36
#define     VersionReg              0x37
#define     AnalogTestReg           0x38
#define     TestDAC1Reg             0x39
#define     TestDAC2Reg             0x3A
#define     TestADCReg              0x3B
#define     Reserved31              0x3C
#define     Reserved32              0x3D
#define     Reserved33              0x3E
#define     Reserved34              0x3F
//-----------------------------------------------

//4 字节卡序列号，第 5 字节为校验字节
uchar serNum[5];
uchar    writeDate[16] ={'T', 'e', 'n', 'g', ' ', 'B', 'o', 0, 0, 0, 0, 0, 0, 0, 0,0};
//扇区 A 密码，16 个扇区，每个扇区密码 6Byte
 uchar sectorKeyA[16][16] = {{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
                             {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
                             {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
                            };
 uchar sectorNewKeyA[16][16] = {{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
                                {0xFF,  0xFF,  0xFF,  0xFF,  0xFF,  0xFF,
0xff,0x07,0x80,0x69, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
                                {0xFF,  0xFF,  0xFF,  0xFF,  0xFF,  0xFF,
0xff,0x07,0x80,0x69, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
                               };
```

```
void setup() {
    Serial.begin(9600);                                  // RFID reader SOUT pin connected to Serial
RX pin at 2400bps
 // start the SPI library:
   SPI.begin();

   pinMode(chipSelectPin,OUTPUT);                        // Set digital pin 10 as OUTPUT to connect
it to the RFID /ENABLE pin
      digitalWrite(chipSelectPin, LOW);                  // Activate the RFID reader
   pinMode(NRSTPD,OUTPUT);                               // Set digital pin 10 , Not Reset and
Power-down
      digitalWrite(NRSTPD, HIGH);

   MFRC522_Init();
}

void loop()
{
      uchar i,tmp;
      uchar status;
            uchar str[MAX_LEN];
            uchar RC_size;
            uchar blockAddr;   //选择操作的块地址 0～63


            //寻卡，返回卡类型
            status = MFRC522_Request(PICC_REQIDL, str);
            if (status == MI_OK)
            {
            }

            //防冲撞，返回卡的序列号 4 字节
            status = MFRC522_Anticoll(str);
            memcpy(serNum, str, 5);
            if (status == MI_OK)
            {
                              Serial.println("The card's number is    : ");
                  Serial.print(serNum[0],BIN);
                  Serial.print(serNum[1],BIN);
                  Serial.print(serNum[2],BIN);
                  Serial.print(serNum[3],BIN);
                  Serial.print(serNum[4],BIN);
                              Serial.println(" ");
            }
```

```
        //选卡，返回卡容量
        RC_size = MFRC522_SelectTag(serNum);
        if (RC_size != 0)
        {}

        //写数据卡
        blockAddr = 7;              //数据块 7
        status = MFRC522_Auth(PICC_AUTHENT1A, blockAddr, sectorKeyA[blockAddr/4],
serNum); //认证
        if (status == MI_OK)
        {
            //写数据
            status = MFRC522_Write(blockAddr, sectorNewKeyA[blockAddr/4]);
                        Serial.print("set the new card password, and can modify the data of
the Sector: ");
                        Serial.print(blockAddr/4,DEC);

                        //写数据
                        blockAddr = blockAddr - 3 ;
                        status = MFRC522_Write(blockAddr, writeDate);
                        if(status == MI_OK)
                        {
                            Serial.println("OK!");
                        }
        }

        //读卡
        blockAddr = 7;              //数据块 7
        status          =          MFRC522_Auth(PICC_AUTHENT1A,          blockAddr,
sectorNewKeyA[blockAddr/4], serNum);     //认证
        if (status == MI_OK)
        {
            //读数据
                        blockAddr = blockAddr - 3 ;
                        status = MFRC522_Read(blockAddr, str);
            if (status == MI_OK)
            {
                                Serial.println("Read from the card ,the data is : ");
                for (i=0; i<16; i++)
                {
                                Serial.print(str[i]);
                }
                                Serial.println(" ");
```

```
                }
            }
                    Serial.println(" ");
            MFRC522_Halt();                 //命令卡片进入休眠状态

}

/*
 * 函 数 名：Write_MFRC5200
 * 功能描述：向 MFRC522 的某一寄存器写一个字节数据
 * 输入参数：addr--寄存器地址；val--要写入的值
 * 返 回 值：无
 */

……
```

因数据边幅很长，所以代码，在这里不全部显示了。

本实验，当 IC 卡靠近后，RFID 模块将写入数据到 IC 卡，然后 RFID 模块再从 IC 卡读出数据，并显示在监控窗口中。