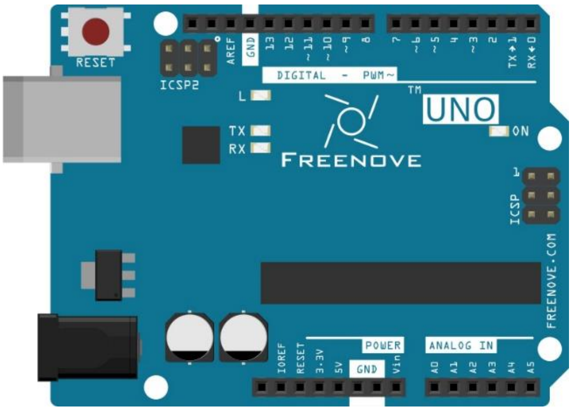
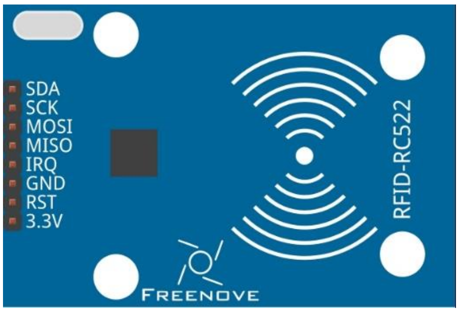
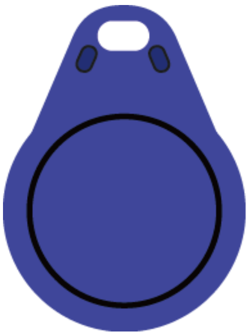
Contents

RFID

RFID（Radio Frequency Identification）is a wireless communication technology. A complete RFID system is generally composed of the responder and reader. Generally, we use tags as responders, and each tag has a unique code, which is attached to the object to identify the target object. The reader is a device for reading (or writing) tag information.
Products derived from RFID technology can be divided into three categories: passive RFID products, active RFID products and semi active RFID products. And Passive RFID products are the earliest, the most mature and most widely used products in the market among others. It can be seen everywhere in our daily life such as, the bus card, dining card, bank card, hotel access cards, etc., and all of these belong to close-range contact recognition. The main operating frequency of Passive RFID products are: 125KHZ (low frequency), 13.56MHZ (high frequency), 433MHZ (ultrahigh frequency), 915MHZ (ultrahigh frequency). Active and semi active RFID products work at higher frequencies.
The RFID module we use is a passive RFID product with the operating frequency of 13.56MHz.

Component List

Freenove UNO x1  	RFID Module （RC522）x1 
Mifare1 S50 Standard card x1 	Mifare1 S50 Non-standard card x1 

USB able x1	Jumper M/F x7
	

Component Knowledge

The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56MHz. The MFRC522 ’s internal transmitter is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443 A/MIFARE cards and transponders without additional active circuitry. The receiver module provides a robust and efficient implementation for demodulating and decoding signals from ISO/IEC 14443 A/MIFARE compatible cards and transponders. The digital module manages the complete ISO/IEC 14443A framing and error detection (parity and CRC) functionality

This RFID Module uses MFRC522 as the control chip, and SPI (Peripheral Interface Serial) as the reserved interface.

Technical specs

Operating Voltage：13-26mA(DC)\3.3V
Idle current：10-13mA(DC)\3.3V
Sleep current in the：<80uA
Peak current：<30mA
Operating frequency：13.56MHz
Supported card type：Mifare1 S50、Mifare1 S70、Mifare Ultralight、Mifare Pro、Mifare Desfire
Physical characteristics：
Size：40mm×60mm
Operation temperature：20-80 degrees(Celsius)
Storage temperature： 40-85 degrees (Celsius)
Operation humidity：5%-95%(Relative humidity)

Mifare1 S50 Card

Mifare S50 is often called Mifare Standard with the capacity of 1K bytes. And each card has a 4-bytes global unique identifier number (USN/UID), which can be rewritten 100 thousand times and read infinite times. Its storage period can last for 10 years. The ordinary Mifare1 S50 Card and non-standard Mifare1 S50 Card equipped for Freenove RFID Kit are shown below.

The Mifare S50 capacity (1K byte) is divided into 16 sectors (Sector0-Sector15). Each sector contains 4 data block (Block0-Block3. 64 blocks of 16 sectors will be numbered according absolute address, from 0 to 63). And each block contains 16 bytes (Byte0-Byte15), 64*16=1024. As is shown in the following table:

Sector No.	Block No.	Storage area	Block type	Absolute block No.
sector 0	block 0	vendor code	vendor block	0
	block 1		data block	1
	block 2		data block	2
	block 3	Password A-access control-password B	control block	3
sector 1	block 0		data block	4
	block 1		data block	5
	block 2		data block	6
	block 3	Password A-access control-password B	control block	7
……	……	……	……	
sector 0	block 0		data block	60
	block 1		data block	61
	block 2		data block	62
	block 3	Password A-access control-password B	control block	63

Each sector has a set of independent password and access control which are put in the last block of each sector, and the block is also known as sector trailer, that is Block 3 in each sector. Sector 0, block 0 (namely absolute address 0) of S50 is used to store the vendor code, which has been solidified and can ’t be changed, and the card serial number is stored here. In addition to the manufacturer and the control block, the rest of the cards are data blocks, which can be used to store data. Data block can be used for two kinds of applications:
(1) used as general data storage and can be operated for reading and writing.
(2) used as data value, and can be operated for initializing the value, adding value, subtracting and reading the value.
The sector trailer block in each sector is the control block, including a 6-byte password A, 4-byte access control and 6-byte password B. For example, the control block of a brand new card is as follows:

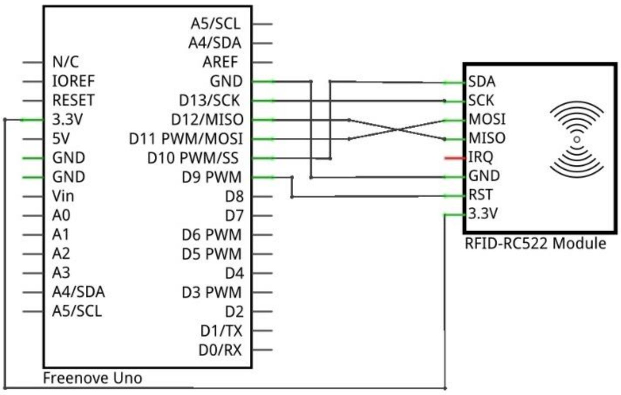
A0 A1 A2 A3 A4 A5	FF 07 80 69	B0 B1 B2 B3 B4 B5
password A	access control	password B

The default password of a brand new card is generally 0A1A2A3A4A5 for password A, B0B1B2B3B4B5 for password B, or both the password A and password B are 6 FF. Access control is used to set the access conditions for each block (including the control block itself) in a sector. Blocks of S50 are divided into data blocks and control blocks. There are four operations, "read", "write", "add value", "subtract value (including transmission and storage)" for data blocks, and there are two operations, "read" and "write" for control blocks.
For more details about how to set data blocks and control blocks, please refer to Datasheet.
By default, after verifying password A or password B, we can do reading or writing operation to data blocks. And after verifying password A, we can do reading or writing operation to control

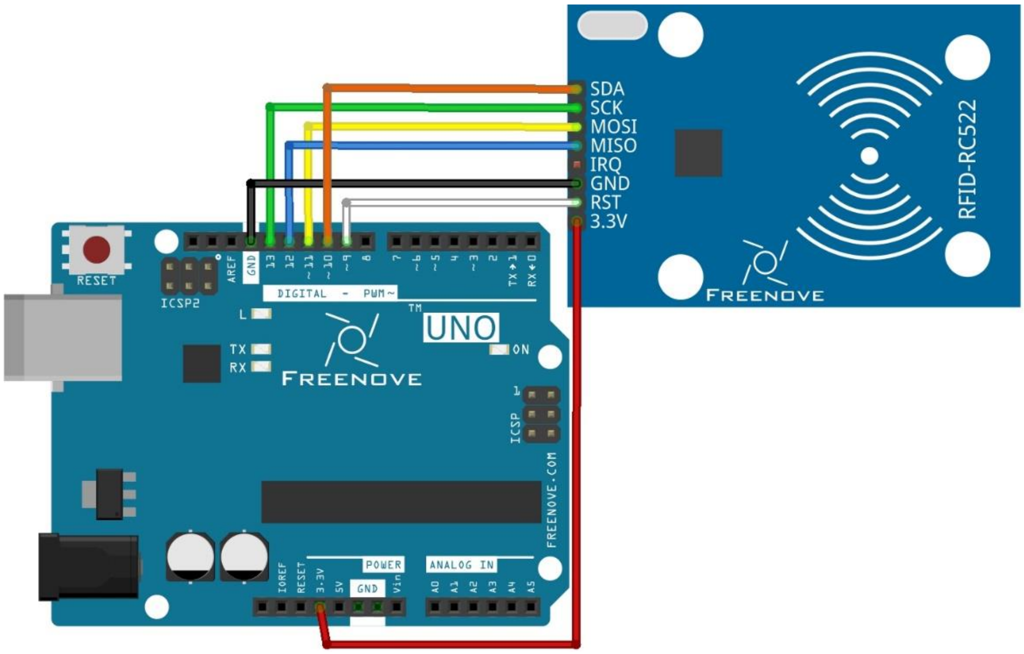
blocks. But password A can never be read. If you choose to verify password A and then you forget the password A, the block will never be able to read again. **It is highly recommended that beginners should not try to change the contents of control blocks.**
For Mifare1 S50 card equipped for Freenove RFID Kit, the default password A and B is FFFFFFFFFFFF. After understanding the above knowledge, we read the unique ID number (UID) of M1S50 card in the Skech1 first. And then do reading and writing operations to the card in the Skech2.

Circuit

Schematic diagram：



Hardware connection：



Sketch 1 Read_UID

Before writing the code, we need import the library RFID.zip first (for the importing method, please refer to the tutorial of Freenove Starter Kit for Arduino). This sketch will read the unique ID number (UID) of the card, recognize the type of the card and display the information through serial port.

```
#include <SPI.h>
#include <RFID.h>

//D10:pin of card reader SDA.  D9:pin of card reader RST
RFID rfid(10,9);
unsigned char status;
unsigned char str[MAX_LEN];  //MAX_LEN is 16: size of the array

void setup()
{
  Serial.begin(9600);
  SPI.begin();
  rfid.init(); //initialization
  Serial.println("Please put the card to the induction area...");
}

void loop()
{
  //Search card, return card types
  if (rfid.findCard(PICC_REQIDL, str) == MI_OK) {
    Serial.println("Find the card!");
    // Show card type
    ShowCardType(str);
    //Anti-collision detection, read card serial number
    if (rfid.anticoll(str) == MI_OK) {
      Serial.print("The card's number is  : ");
      //Display card serial number
      for(int i = 0; i < 4; i++){
        Serial.print(0x0F & (str[i] >> 4),HEX);
        Serial.print(0x0F & str[i],HEX);
      }
      Serial.println("");
    }
    //card selection (lock card to prevent redundant read, removing the line will make the sketch read cards continually)
```

```
      rfid.selectTag(str);
    }
    rfid.halt();  // command the card to enter sleeping state
}
void ShowCardType(unsigned char * type)
{
    Serial.print("Card type: ");
    if(type[0]==0x04&&type[1]==0x00)
      Serial.println("MFOne-S50");
    else if(type[0]==0x02&&type[1]==0x00)
      Serial.println("MFOne-S70");
    else if(type[0]==0x44&&type[1]==0x00)
      Serial.println("MF-UltraLight");
    else if(type[0]==0x08&&type[1]==0x00)
      Serial.println("MF-Pro");
    else if(type[0]==0x44&&type[1]==0x03)
      Serial.println("MF Desire");
    else
      Serial.println("Unknown");
}
```

After including the RFID library, we need to construct a RFID class object before using the function in RFID library. Its constructor needs to be written to two pins, respectively to the SDA pin and the RST pin.

```
RFID rfid(10,9);
```

In setup, initialize the serial port, SPI and RFID.

```
  Serial.begin(9600);
  SPI.begin();
  rfid.init(); //initialization
```

In loop (), use .findCard () waiting for the card approaching. Once it detects card contact, this function will return MI_OK and save the card type data in parameter str. Then enter the if statement.

```
if (rfid.findCard(PICC_REQIDL, str) == MI_OK)
```

After entering if statement, call the sub function ShowCardType(). Then determine the type of the card according to the content of STR and print the type out through the serial port.

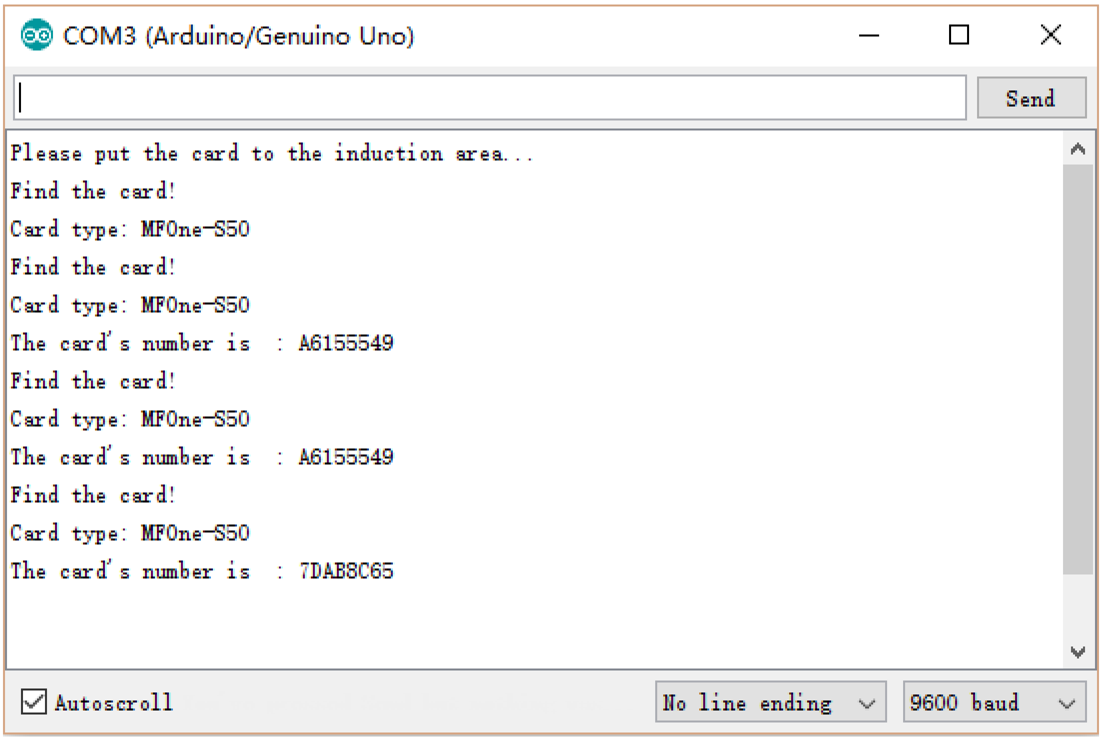
```
ShowCardType(str);
```

Then use the.anticoll() to read UID of the card and use serial port to print it out.

```
    if (rfid.anticoll(str) == MI_OK) {
      Serial.print("The card's number is  : ");
      // Display serial number of card
      for(int i = 0; i < 4; i++){
        Serial.print(0x0F & (str[i] >> 4),HEX);
```

```
            Serial.print(0x0F & str[i],HEX);
        }
        Serial.println("");
    }
```

After verifying and uploading the code to UNO, open the serial port monitor and make a card approach the sensing area of RFID module. Then serial port monitoring window will display the displacement ID number and the type of the card. If the induction time is too short, it may lead to incomplete-information display.



Sketch 2 ReadAndWrite

In this part, first read the data in particular location of the S50 M1 Card, then write data in that position and read it out. Display these data through the serial port.

```
#include <SPI.h>
#include <RFID.h>

// D10:pin of card reader SDA.  D9: pin of card reader RST
RFID rfid(10, 9);

// 4-byte card serial number, the fifth byte is check byte
unsigned char serNum[5];
unsigned char status;
unsigned char str[MAX_LEN];
unsigned char blockAddr;        //Select the operation block address: 0 to 63
```

```

// Write card data you want(within 16 bytes)
unsigned char writeDate[16] = "WelcomeFreenove";

// The A password of original sector: 16 sectors; the length of password in each sector
is 6 bytes.
unsigned char sectorKeyA[16][16] = {
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
  { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ,
};

void setup()
{
  Serial.begin(9600);
  SPI.begin();
  rfid.init();
  Serial.println("Please put the card to the induction area...");
}

void loop()
{
  //find the card
  rfid.findCard(PICC_REQIDL, str);
  //Anti-collision detection, read serial number of card
  if (rfid.anticoll(str) == MI_OK) {

    Serial.print("The card's number is  : ");
    //print the card serial number
```

```
    for (int i = 0; i < 4; i++) {
      Serial.print(0x0F & (str[i] >> 4), HEX);
      Serial.print(0x0F & str[i], HEX);
    }
    Serial.println("");
    memcpy(rfid.serNum, str, 5);
  }
  //select card and return card capacity (lock the card to prevent multiple read and
written)
  rfid.selectTag(rfid.serNum);
  //first, read the data of data block 4
  readCard(4);
  //write data(within 16 bytes) to data block
  writeCard(4);
  //then read the data of data block again
  readCard(4);

  rfid.halt();
}
//write the card
void writeCard(int blockAddr) {

  if (rfid.auth(PICC_AUTHENT1A, blockAddr, sectorKeyA[blockAddr / 4], rfid.serNum) ==
MI_OK) //authenticate
  {
    //write data
    //status = rfid.write((blockAddr/4 + 3*(blockAddr/4+1)), sectorKeyA[0]);
    Serial.print("set the new card password, and can modify the data of the Sector: ");
    Serial.println(blockAddr / 4, DEC);
    // select block of the sector to write data
    if (rfid.write(blockAddr, writeDate) == MI_OK) {
      Serial.println("Write card OK!");
    }
  }
}
//read the card
void readCard(int blockAddr) {

  if ( rfid.auth(PICC_AUTHENT1A, blockAddr, sectorKeyA[blockAddr / 4], rfid.serNum) ==
MI_OK) // authenticate
  {
    // select a block of the sector to read its data
```

```
    Serial.print("Read from the blockAddr of the card : ");
    Serial.println(blockAddr, DEC);
    if ( rfid.read(blockAddr, str) == MI_OK) {
      Serial.print("The data is : ");
      Serial.println((char *)str);
    }
  }
}
```

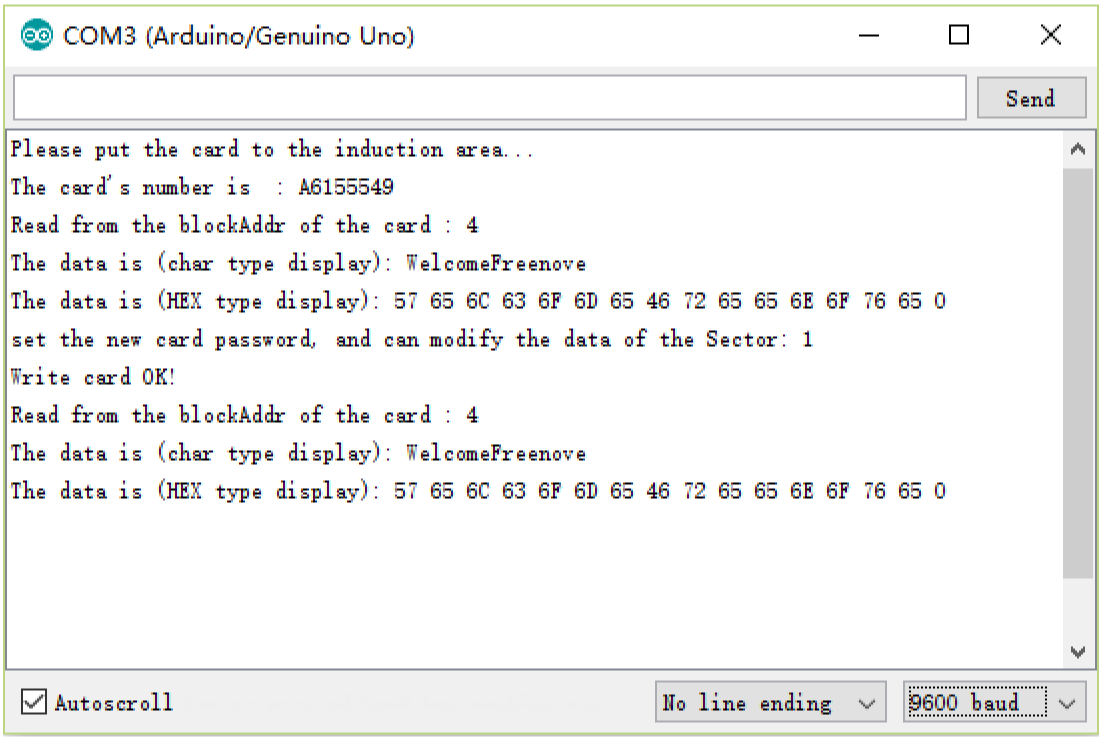
In the sub function of writeCard () and readCard (), we must first verify the password A, and then use the corresponding sub function to read and write. Here we do reading and writing operations to data block 0 (absolute NO.4) of the first sector.

```
if (rfid.auth(PICC_AUTHENT1A, blockAddr, sectorKeyA[blockAddr / 4], rfid.serNum) == MI_OK) //认证 Approve
  {……
```

In loop (), compare the contents of the data block NO.4 after written to the original contents.

```
  // first, read the data of data block 4
  readCard(4);
  // write data(within 16 bytes) to data block
  writeCard(4);
  // then read the data of data block again
  readCard(4);
```

After verifying and uploading the code to UNO, open the serial port monitor and make a card approach the sensing area of RFID module, then the serial port monitoring window will display displacement ID numbers of the card, the type of this card and the contents (before and after writing operation) of data block. If the induction time is too short, it may lead to incomplete information display.


COM3 (Arduino/Genuino Uno)
Send
Please put the card to the induction area...
The card's number is : A6155549
Read from the blockAddr of the card : 4
The data is (char type display): WelcomeFreenove
The data is (HEX type display): 57 65 6C 63 6F 6D 65 46 72 65 65 6E 6F 76 65 0
set the new card password, and can modify the data of the Sector: 1
Write card OK!
Read from the blockAddr of the card : 4
The data is (char type display): WelcomeFreenove
The data is (HEX type display): 57 65 6C 63 6F 6D 65 46 72 65 65 6E 6F 76 65 0
Autoscroll
No line ending
9600 baud