

Считыватель RFID RC522 13.56MHz

Считыватель RFID RC522 – периферийное устройство более сложного прибора. Радиоидентификация RFID (Radio Frequency Identification) происходит при обмене данными по протоколу Mifare 1K. Использовать термин Mifare может только компания NXP Semiconductors, а также компании, имеющие лицензию от NXP на производство чипов (сейчас компания Infineon). Эта технология позволяет автоматически опознавать объекты, содержащие RFID метки – так называемые транспондеры. Из меток данные могут не только читаться, но и записываться.

Mifare – торговая марка, объединяющая несколько типов микросхем пластиковых карт, микросхемы считывания и записи стационарных приборов и различные продукты на их основе. Продукты Mifare соответствуют стандарту ISO 14443 Type A пластиковых карт. Перевод стандарта на русский язык ГОСТ Р ИСО/МЭК 14443. Обмен данными по радио происходит через рамочные антенны, находящиеся в карточке и в модуле. Сигнал модуля служит источником энергии для метки. Считыватель RFID RC522 срабатывает при поднесении метки. Основа модуля – микросхема MFRC522. Он может обрабатывать информацию одновременно от нескольких меток. Для считывания информации достаточно ненадолго попасть карте в зону регистрации даже при перемещении на большой скорости.

В наше время происходит бурное внедрение RFID технологий в различных областях жизни. RFID системы используются для оплаты общественного транспорта в Москве, Санкт-Петербурге, Казани, Минске, Баку и в других городах. Вошли в употребление бесконтактные проездные билеты метро, представляющие собой карточку с RFID меткой. Зародившись как альтернатива жетонам и бумажным билетам общественного транспорта, пропускам, всевозможным талонам, радиоидентификация шагнула в складской учет и автоматизацию конвейерного производства. Нашлось применение карточек для малых платежей в школьных и студенческих столовых. Но по-прежнему главным применением RFID является опознавание своей, чужой.

Характеристики считывателя RFID RC522

Питание:

- Напряжение 3,3 В
- Ток потребления в режимах
 - дежурный 80 мкА
 - ожидания 12 мА
 - обычный не более 26 мА
 - наибольший 30 мА

Частота HF 13,56 МГц

Частотная полоса 13,55–13,57 МГц

Расстояние считывания 0–25 мм

Сопровождаемые карты:

- классы S50, S70, Ultralight, Pro, DESFire
- ипы Mifare S50, Mifare S70, Mifare UltraLight, Mifare Pro, Mifare DESfire

Скорость передачи информации 106, 212, 424, 848 кбит/с

Стандарт протокола NFC Reader ISO 14443 A Mifare classic protocol

Шифрование Security Features Mifare classic™

Размеры 40 x 60 мм

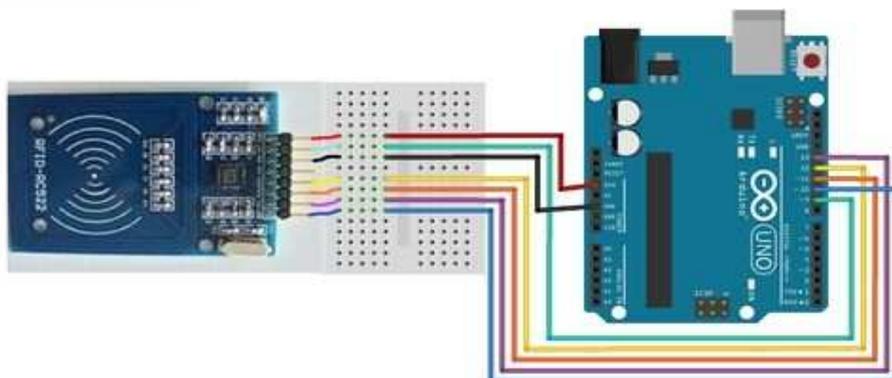
Температура:

- рабочая –20...80 С°
- хранения –40...85 С°

Относительная влажность 5–95 %

Контакты и сигналы RFID RC522

Сигнал сброса, поступающий на контакт RST считывателя это не сброс Arduino или Raspberry Pi. Это совершенно другой сигнал. Он должен поступать от цифрового выхода МК и формируется программно. При подаче логического 0 происходит перезагрузка считывателя. Также считыватель сообщает низким уровнем на RST, что находится в режиме сна, и чтоб его "разбудить" необходимо туда же подать высокий уровень.



Подключение к Arduino Uno.

Подключение RFID RC522 к различным типам контроллера Arduino.

RFID rc522	Arduino Mega	Arduino Uno и Nano v3	Arduino Leonardo и Micro	Arduino Pro Micro	CraftDuino	Raspberry Pi B+
MISO	50	12	ICSP-1	14	12	21
MOSI	51	11	ICSP-4	16	11	19
SCK	52	13	ICSP-3	15	13	23
SDA	53		10			24

Контакт главного модуля для подключения к контакту SDA указывается как SS_PIN в программе. Контакт главного модуля для подключения к контакту RST указывается как RST_PIN в программе. Это выполняется с помощью команд:

```
#define SS_PIN номер контакта
```

и

```
#define RST_PIN номер контакта
```

Все контакты модуля кроме IRQ обязательно подключаются. Сигнал от контакта IRQ обрабатывается программно.

Запись с помощью считывателя RFID RC522

Устройство может применяться как программатор карточек. С его помощью можно не только читать хранящиеся данные, корректировать используя ПК и записать вновь, но и изменить идентификационный код.

Пластиковая карта

В комплект входит белая пластиковая карта. Mifare 1K на которую можно нанести цветное изображение. Внутри нее находятся антенна и микросхема Mifare S50, содержащая память и

радиочасть. Размер памяти 1 килобайт, тип EEPROM. Она разделена на 16 секторов, состоящих из 4 разделов. В каждом разделе три информационных части и одна для ключей. Внутри одной части есть 16 байт памяти. Срок хранения данных 10 лет, количество циклов перезаписи 100000. Уникальность карточки Mifare обеспечивается присвоением изготовителем номера. Он используется в качестве идентификационного кода. Для защиты хранящихся данных в микросхеме карты использовано аппаратное шифрование. При работе данные с пластиковой карточки поступают на считыватель только после взаимной идентификации кода, записанного в сектор памяти карточки и хранящегося в считывателе.

Характеристики карты

Тип Mifare Standard 1k (тонкая)

Время транзакции 0,164 с

Температура:

-рабочая –30...75 C°

-хранения –40...85 C°

Размеры: 86 x 54 x 0,8 мм

Определение номера карты

Для записи и чтения с карты необходимо знать ее уникальный номер, необходимый для работы системы радиоидентификации. Определить номер можно используя программу Arduino. Соедините считыватель RFID RC522 и Arduino UNO. Воспользуйтесь готовой библиотекой, установив ее в Arduino IDE с помощью копирования в соответствующую папку. Запишите программу.

```
#include
```

```
#include "RFID.h"
```

```
#define SS_PIN 10
```

```
#define RST_PIN 9
```

```
RFID rfid(SS_PIN, RST_PIN);
```

```
// Setup variables:
```

```
int serNum0;
```

```
int serNum1;
```

```
int serNum2;
```

```
int serNum3;
```

```
int serNum4;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  SPI.begin();
```

```
  Serial.println("12");
```

```
  rfid.init();
```

```
}
```

```
void loop()
```

```
{
```

```
  if (rfid.isCard()) {
```

```
    if (rfid.readCardSerial()) {
```

```
      if (rfid.serNum[0] != serNum0
```

```
        && rfid.serNum[1] != serNum1
```

```

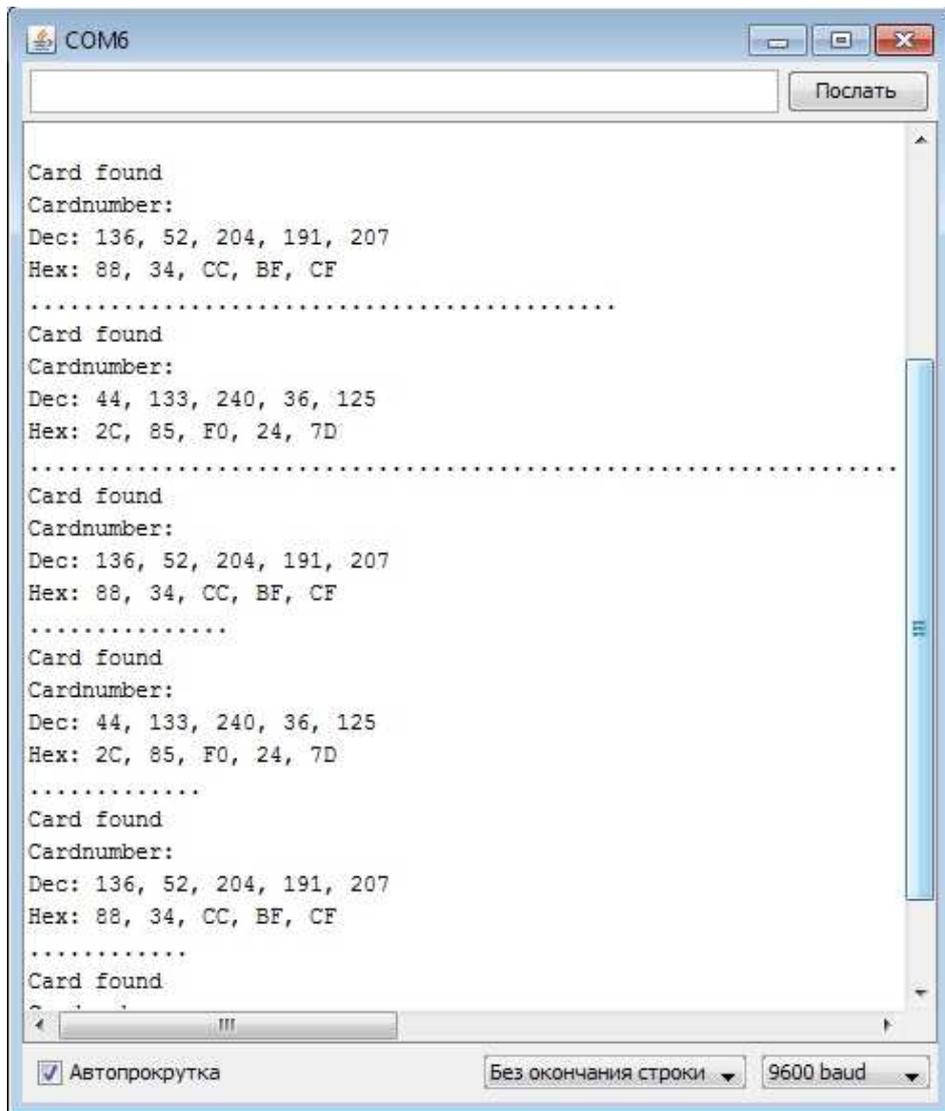
&& rfid.serNum[2] != serNum2
&& rfid.serNum[3] != serNum3
&& rfid.serNum[4] != serNum4
) {
/* With a new cardnumber, show it. */
Serial.println(" ");
Serial.println("Card found");
serNum0 = rfid.serNum[0];
serNum1 = rfid.serNum[1];
serNum2 = rfid.serNum[2];
serNum3 = rfid.serNum[3];
serNum4 = rfid.serNum[4];

//Serial.println(" ");
Serial.println("Cardnumber:");
Serial.print("Dec: ");
Serial.print(rfid.serNum[0],DEC);
Serial.print(", ");
Serial.print(rfid.serNum[1],DEC);
Serial.print(", ");
Serial.print(rfid.serNum[2],DEC);
Serial.print(", ");
Serial.print(rfid.serNum[3],DEC);
Serial.print(", ");
Serial.print(rfid.serNum[4],DEC);
Serial.println(" ");

Serial.print("Hex: ");
Serial.print(rfid.serNum[0],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[1],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[2],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[3],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[4],HEX);
Serial.println(" ");
} else {
/* If we have the same ID, just write a dot. */
Serial.print(".");
}
}
}

rfid.halt();
}

```



Программа выводит ряд чисел: 44, 133, 240, 36, 125. Пишем их в обратном порядке. Убираем первое число, (контрольная сумма, оно только что было последним) и оставшиеся числа переводим в шестнадцатеричный вид. Пишем в том же порядке но без пробелов. Теперь это большое число переводим в десятичный вид и получаем номер карты.