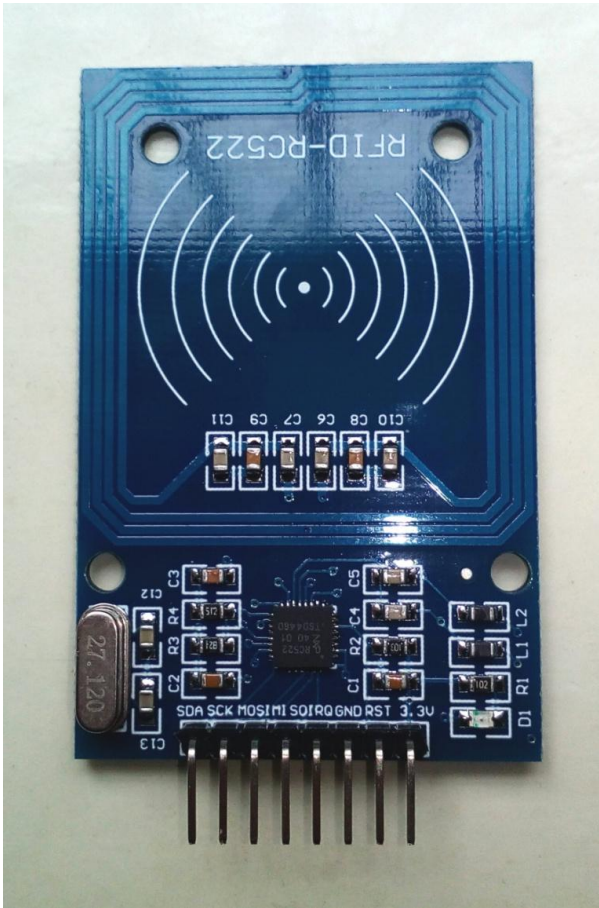


36 – Striscia la scheda ed aprì la porta - Slide card to open a door

(some notes at end of this section)



RFID (Radio Frequency Identification) e' un sistema di trasferimento dati su breve distanza (normalmente meno 20 cm) una cui importante caratteristica e' quella di poter alimentare solo uno dei due terminali. Questa caratteristica fa si che il terminale non alimentato (il terminale passivo) possa essere utilizzato in situazioni per le quali e' impossibile fornire energia come, ad esempio, nel chip identificativo dei cani (inserito sottopelle) oppure nei sistemi bloccamotore, inseriti nelle chiavi delle automobili, nei sistemi di pagamento contactless, nei bracciali che spesso ci vengono forniti per entrare nelle piscine e, una sua variante, anche come "sistema anti taccheggio" in un'infinita' di prodotti di abbigliamento e non, in vendita nei supermercati. L'etichetta che siamo costretti a tagliare dagli indumenti appena acquistati o la strana placchetta che qualche volta troviamo sotto l'etichetta di vari generi alimentari altro non sono che terminali RFID, inseriti al solo scopo di far scattare un allarme nel momento in cui incrociano l'altro terminale (normalmente le porte a fianco delle casse), debitamente alimentato.

In figura vediamo il modulo alimentato (o terminale attivo), in grado di leggere e modificare i dati di un terminale passivo. Si presenta come una scheda elettronica con una vasta area (l'antenna) per la "cattura" del segnale. A fianco, sottoforma di portachiavi, il modulo non alimentato. Esistono numerosi tipi di moduli passivi, dai piu' semplici, in grado solo di segnalare la loro presenza (normalmente usati nei sistemi antitaccheggio) a quelli piu' complessi, dotati non solo di un numero identificativo (unico per ogni singolo modulo passivo) ma anche di memorie nelle quali possono essere inseriti dati e storia dell'oggetto cui sono associati. Esistono anche moduli passivi alimentati (non trattati in queste note), in grado di raccogliere e memorizzare informazioni anche in assenza di un modulo attivo.

Il modulo RC522 in figura e' un modulo piuttosto complesso a bordo del quale sono presenti parecchi componenti (l'antenna, ovviamente, ma anche led, resistenze e condensatori, oltre ad un

Arduino: RFID RC522: striscia la scheda ed apri la porta - slide card to open a door

oscillatore quarzato (per la gestione della frequenza di trasmissione) ed un chip in grado di elaborare i segnali raccolti dall'antenna e di trasformarli in impulsi digitali, leggibili da Arduino tramite una specifica libreria.

Contrariamente a quel che normalmente accade per altri componenti, i pin del modulo RC522 devono essere collegati ad alcune porte fisse di Arduino (11, 12 e 13 per Arduino uno R3, altri pin per altri modelli di Arduino) poiché la loro posizione è predefinita nella libreria di utilizzo. Questa rigidità, unita alla necessità di utilizzare almeno 5 porte, pone qualche limite all'utilizzo di questo modulo su Arduino uno, il cui numero di porte è notoriamente limitato.

In questo esercizio utilizzeremo schede RFID del tipo MIFARE classic da 1 kbyte. Queste schede non solo sono dotate di un codice identificativo ma dispongono di sedici blocchi di memoria in ognuno dei quali possono essere inseriti un massimo di 47 byte. Non molto, evidentemente, ma sufficienti a memorizzare date ed informazioni relative all'oggetto al quale la scheda è associata.

Il modulo attivo viene utilizzato solo come lettore di uno o più moduli passivi (di una o più schede). Il lettore legge il codice del modulo passivo, lo visualizza sul monitor seriale e, se identico a quello memorizzato in una tabella interna al programma, accende un led verde e apre una porta pilotata da un servomotore. In caso contrario (se cioè il modulo RFID ha un codice non memorizzato in tabella), fa lampeggiare un led rosso e lancia un allarme di cinque secondi.

Una specie di porta di sicurezza, quindi, apribile solo se si è in possesso del badge con il giusto codice RFID. L'esercizio utilizza, per ragioni pratiche, un servomotore SG90 per simulare l'apertura di una porta, ma nulla osta che al posto del servomotore possa essere inserito un relè in grado di lanciare l'impulso di apertura di una serratura elettrica.

Prima di procedere alla compilazione dello sketch è necessario installare la libreria di gestione del modulo RC522:

- scaricare la cartella in formato zip presente a questo indirizzo:
<https://github.com/miguelbalboa/rfid> (premere il bottone "download zip" presente sul lato destro dello schermo)
- aggiungere la libreria zippata al repository dell'IDE (IDE -> sketch -> include library -> add zip library)

Prima di procedere all'utilizzo del programma bisogna "specializzarlo" con il codice identificativo della scheda valida. Per fare questo si deve:

- compilare il programma e trasferirlo su Arduino;
- aprire il monitor seriale (ctrl + maiusc + M) oppure strumenti -> monitor seriale;
- strisciare il portachiavi o la tessera che si vuole utilizzare;
- leggere sul monitor seriale il codice identificativo della portachiavi o della tessera (quattro numeri)
- inserire i quattro numeri all'interno delle parentesi graffe in riga 30 del programma (al posto di: 232, 118, 45, 0); questa la riga da modificare: `char(codicevalido [5]) = {232, 118, 45, 0};`
- ricompilare e provare il programma.

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi è anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it



Here some notes about this project, translated by google translator

RFID (Radio Frequency Identification) is a short-distance data transfer system (usually less than 20 cm) where an important feature is to feed only one of the two terminals. This feature means that the non-powered terminal (the passive terminal) can be used in situations for which is impossible to provide energy such as, for example, in a dog identifier chip (inserted under the skin) or in an engine immobilizer system, inserted in car keys, in contactless payment systems, in bracelets that often are provided to enter the pools, and a variant, also as "shoplifting anti system" on an infinite numbers of clothing products and not, for sale in supermarkets.

The label that we are forced to cut in clothes just bought or the strange label that sometimes we find under the price label of various and expensive foodstuffs, are RFID terminals, inserted in order to trigger an alarm when you cross the active terminal (normally sided on the exit doors).

In the figure we see the powered terminal (active terminal), able to read and modify data on a passive terminal. It presents as an electronic card with a large area (the antenna) to "capture" the signal. Alongside, in the form of key rings the passive, non powered device. There are numerous types of passive devices, from the most simple, able to signal their presence only, to those more complex, that not only have an identification number (unique for each passive module) but also able to memorize data about the object which they are associated. There are also fed passive modules (not treated in these notes), able to collect and store information even in absence of an active device.

The RC522 device in the figure is rather complex. On board there are several components. The antenna, of course, but also LEDs, resistors, capacitors, a quartz oscillator (for the transmission frequency management) and a chip capable of processing the signals collected by antenna and turn them into digital pulses, readable by Arduino.

Contrary to what usually happens with other components, some the pins of the RC522 device must be connected to arduino on fixed pins (11, 12 and 13 for Aduino uno R3, other pins to other Arduino models) because their position is fixed from library. This rigidity, combined with the use at least 5 pins, poses some limitation on use of this device on Arduino uno, due to limited number of pins.

In this example we will use RFID cards Mifare Classic 1 kbyte. These boards not only are equipped with an identification code but they have sixteen memory blocks, in each of which can be inserted a maximum of 47 bytes. Not too much, of course, but enough to store data about the associated object.

The active device is used only as reader of one or more passives devices. The reader reads the code on passive device, displays it on serial monitor and, if identical to one stored in a program internal table, a green LED lights up and opens a door driven by a servomotor. Otherwise (if the passive device has a code not stored in the program internal table), flashes a red light and launches a five-seconds alarm.

A kind of security door, that can be opened only if you own a right code badge. We use, for practical reasons, a SG90 servomotor to simulate a door opening, but you can place, instead of the servomotor, a relay for an electric lock opening.

Before proceeding to program compilation must be installed, if not already done, the library:

- MFRC522.h found [here](#).

For installation, see process shown in previous examples, and summarized in:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation please verify the library. It must be present in IDE-> sketch-> includes Library-> Contributed library

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

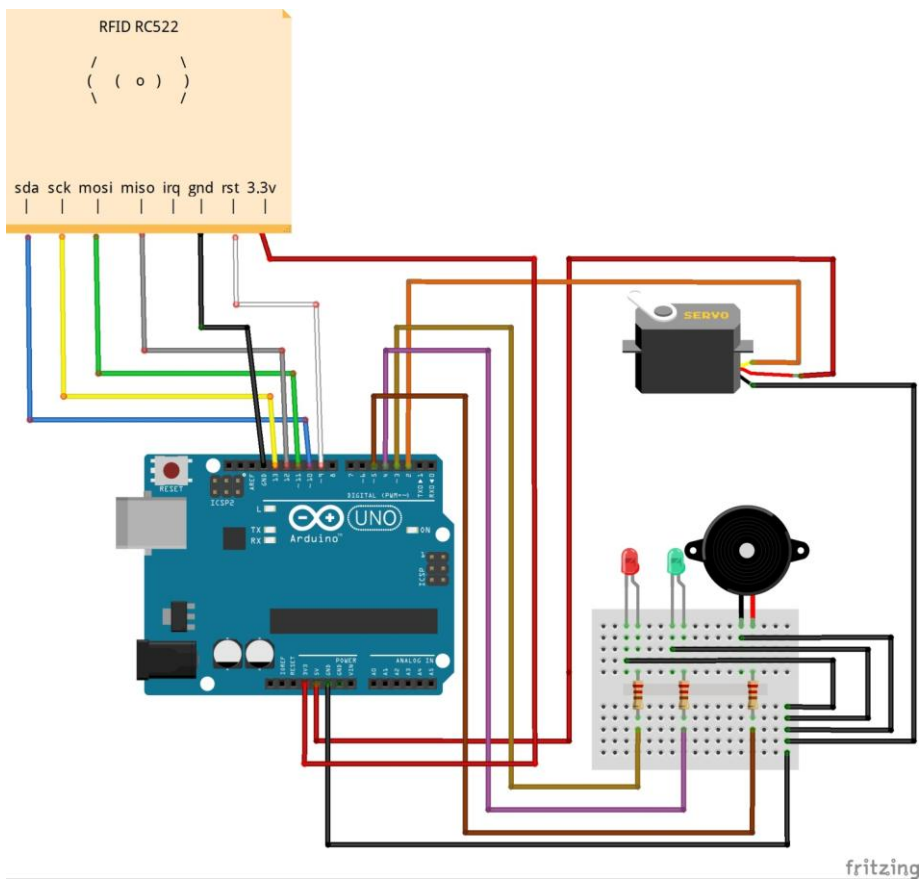
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

Materiali

- Un lettore RFID RC522
- Due o piu' schede MIFARE da 1 k byte
- Due led (uno rosso ed uno verde)
- Un buzzer attivo
- Tre resistenze da 220 ohm
- Un servomotore SG90

Schema



Programma

Download del programma tramite Dropbox
Program download via Dropbox

/ Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il trasferimento nell'IDE, premendo CTRL+T.*

Arduino: RFID RC522: striscia la scheda ed apri la porta - slide card to open a door

Questo programma e' derivato da programmi presenti nella libreria MFRC522.

```
* Per maggiori dettagli sull'utilizzo della scheda RC522 fare riferimento
* alla suddetta libreria, reperibile in https://github.com/miguelbalboa/rfid
*
* Nota: La libreria utilizza in maniera predefinita alcune porte di Arduino
* per cui, per poter ripetere questo esercizio, e' necessario rispettare i
* seguenti collegamenti tra i pin della scheda RC522 e le porte di Arduino:
* -----
* .....pin .....Arduino .....Arduino ...Arduino.....Arduino..... Arduino
* Segnale...MFRC522 ... Uno ..... Mega .....Nano v3...Leonardo/Micro....Pro Micro
* -----
* RST/Reset ...RST ..... 9 ..... 5 ..... D9 .....RESET/ICSP-5 ..... RST
* SPI SS .....SDA(SS)... 10 ..... 53 ..... D10 ..... 10 ..... 10
* SPI MOSI ...MOSI .... 11 ..... 51 ..... D11 .....ICSP-4 ..... 16
* SPI MISO ...MISO .... 12 ..... 50 ..... D12 .....ICSP-1 ..... 14
* SPI SCK ....SCK ..... 13 ..... 52 ..... D13 .....ICSP-3 ..... 15
* -----
*
* -----
* Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
* This program is derived from programs in MFRC522 library.
* Note: The library uses by default, some Arduino pins so, to be able to repeat this exercise, you
* must comply with the above connections between pins of RC522 device and Arduino.
* -----
*/
#include <SPI.h> // richiama la libreria di gestione del bus SPI
#include <MFRC522.h> // richiama la libreria di gestione del lettore rc522
#include <Servo.h> // richiama la libreria di gestione dei servomotori
#define RST_PIN 9
#define SS_PIN 10
#define ledrosso 3
#define ledverde 4
#define buzzer 5
MFRC522 mfrc522(SS_PIN, RST_PIN); // Crea un "oggetto" MFRC522
Servo porta; // crea il servo oggetto "porta"
char codice [5]; // variabile di memorizzazione del codice identificativo
// presente sulla scheda intercettata dal lettore
char(codicevalido [5]) = {232, 118, 45, 0}; // variabile di memorizzazione del codice
// (valore ascii) della scheda valida per aprire la porta
int pos = 0; // variabile di memorizzazione della posizione della porta
int i = 0; // indice utilizzato per scorrere la tabella di memorizzazione
// dei caratteri identificativi della scheda
int b = 0; // indice utilizzato per modulare la frequenza del buzzer
int c = 0; // indice utilizzato per gestire la durata dell'allarme
int semaforo = 0; // semaforo utilizzato per segnalare l'uguaglianza tra il codice
// letto ed il codice valido: 0 = codici non uguali;
// 1 = codici uguali
//
/* ***** routine di apertura della porta ***** */
/*****open door routine *****/
void apreporta(void)
{
    digitalWrite (ledverde, HIGH); // accende il led verde
    digitalWrite (ledrosso, LOW); // spegne il led rosso (qualora fosse acceso)
    digitalWrite (buzzer, HIGH); // attiva il buzzer per un decimo di secondo, per
    // segnalare il consenso
    delay (100);
    digitalWrite (buzzer, LOW);
    //
    /***** gestione della fase di apertura della porta *****/
    for (pos = 0; pos < 90; pos ++) // sposta di un grado per volta l'angolazione della porta
    {
        porta.write(pos); // indirizza il perno alla posizione memorizzata in 'pos'
        delay(30); // attende 30 ms per consentire al servomotore di
        // raggiungere la posizione
    }

    delay (2000); // lascia aperta la porta per due secondi
    //
    /***** gestione della fase di chiusura della porta *****/
    for (pos = 90; pos >= 1; pos--) // diminuisce l'angolazione della porta
    {
        porta.write(pos); // indirizza il perno alla posizione memorizzata in 'pos'
        delay(15); // attende 15ms per consentire al servomotore di
        // raggiungere la posizione
    }

    digitalWrite (ledverde, LOW); // al termine della chiusura, spegne il led verde
```


Arduino: RFID RC522: striscia la scheda ed apri la porta - slide card to open a door

```
}

/* ***** routine di allarme *****
***** alarm routine *****
*/
void lanciaallarme (void)
{
    for (c = 0; c < 4; c++)          // lancia tre cicli di suono bitonale (in tutto circa 5 secondi)
    {
        digitalWrite (ledrosso, HIGH); // accende il led rosso
        digitalWrite (ledverde, LOW); // spegne il led verde, qualora fosse acceso
        for (b = 0; b < 350; b++)      // definisce la durata del primo suono (350 cicli da 2
            //                          millisecondi).
        {
            digitalWrite (buzzer, HIGH); // attiva il suono
            delay(1);                     // attende 1 millisecondo. Di fatto attiva e disattiva ogni 2
            //                          millescondi e quindi 500 volte al minuto
            digitalWrite (buzzer, LOW); // disattiva il suono
            delay(1);                     // attende 1 millisecondo e riparte dall'istruzione for
            //                          (che ripete per 350 volte)
        }

        delay (50);                    // attende 50 millisecondi prima di lanciare la seconda
        //                          sequenza di suoni
        digitalWrite (ledrosso, LOW); // spegne il led
        for (b = 0; b < 150; b++)      // definisce la durata del secondo suono (150 cicli da
            //                          4 millisecondi)
        {
            digitalWrite (buzzer, HIGH); // attiva il suono
            delay(2);                     // attende 2 millisecondi (ripete il suono ogni 4 millisecondi
            //                          quindi 250 volte al minuto
            //e
            digitalWrite(buzzer, LOW); // spegne il suono
            delay(2);                     // attende 2 millisecondi e ripete il ciclo per 150 volte
        }
    }
}

//
void setup()
{
    porta.attach(2);                  // assegna il servo oggetto "pippo" alla porta 2
    Serial.begin(9600);               // Inizializza la comunicazione seriale con il pc
    SPI.begin();                     // Inizializza il bus SPI
    mfrc522.PCD_Init();              // Inizializza il lettore MFRC522
    Serial.print("codice della scheda valida: ");
    Serial.println (codicevalido); // visualizza il codice (in caratteri) della scheda valida
    pinMode (ledrosso, OUTPUT); // definisce la porta 3 (led rosso) come porta di output
    pinMode (ledverde, OUTPUT); // definisce la porta 4 (led verde) come porta di output
    pinMode (buzzer, OUTPUT); // definisce la porta 5 (buzzer) come porta di output
    porta.write(0);                  // indirizza il perno alla posizione 0 (chiude la porta,
    //                          qualora fosse aperta)
    delay (500);                     // attende mezzo secondo per consentire al perno di
    //                          raggiungere la posizione zero
}

void loop()
{
    if ( ! mfrc522.PICC_IsNewCardPresent()) // verifica se una scheda e' vicina al lettore
    {
        return;                          // ricomincia il loop se non c'e' nulla vicino al lettore
    }

    if ( ! mfrc522.PICC_ReadCardSerial()) // se intercetta un modulo rfid, verifica se si tratta
    // di modulo RC522
    {
        return;                          // ricomincia il loop se il modulo intercettato non e' RC522
    }

    Serial.print ("letta scheda con codice identificativo: ");
    for (i = 0; i < mfrc522.uid.size; i++) // acquisisce, tramite le funzioni della libreria,
    // l'identificativo della scheda
    {
        Serial.print (mfrc522.uid.uidByte[i]); // visualizza il codice della scheda letta (valore ascii
        // dei quattro caratteri identificativi della scheda)
        Serial.print (" ");
        codice [i] = mfrc522.uid.uidByte[i]; // memorizza il carattere appena letto del codice
    }
}
```

Arduino: RFID RC522: striscia la scheda ed apri la porta - slide card to open a door

```
//                                della scheda
}

Serial.print ("  ascii: ");
Serial.println (codice);           // visualizza il codice sottoforma di caratteri
semaforo = 1;                      // pone preventivamente a 1 il semaforo
for (i = 0; i < 5; i++)            // verifica, carattere per carattere, se il codice letto
//                                corrisponde al codice valido memorizzato
{
  if (! (codice [i] == codicevalido [i])) // se il carattere letto non corrisponde al carattere
//                                valido memorizzato
    semaforo = 0;                  // spegne il semaforo (indica che codice letto non e' valido)
}

if (semaforo == 1)                // verifica il valore del semaforo
{ // se il semaforo vale 1 (se il codice letto e' uguale al codice memorizzato)
  apreporta();                    // lancia la routine di apertura della porta
}

else                               // in caso contrario
{ // se il codice non e' valido
  lanciallarme();                // lancia la routine di allarme
}
}
```