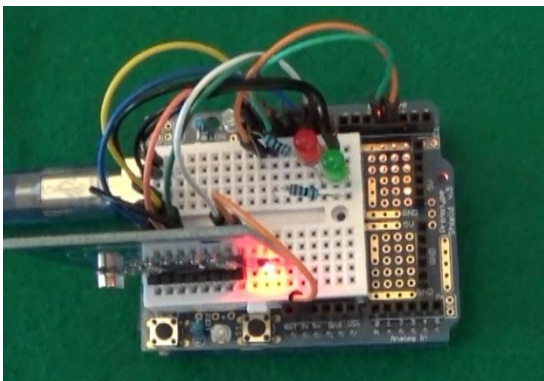
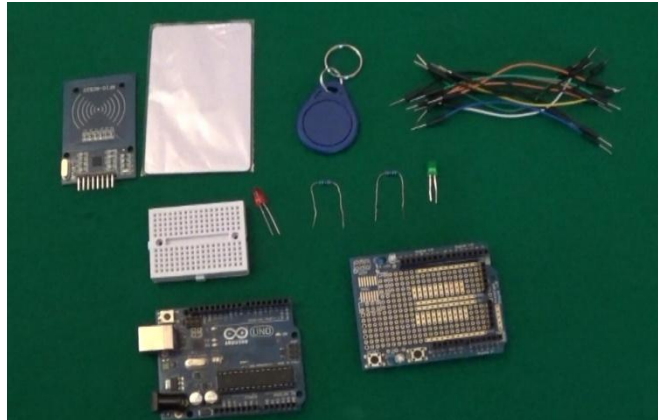


36bis RFID: Azioni diverse per ogni scheda - RFID: different action for each card (some notes at end of this section)

Alcune indicazioni sui principi e sulle caratteristiche dei sistemi RFID sono presenti nella scheda dell'esercizio 36 ([premi qui' per scaricare la scheda](#)) per cui in questa scheda non si parla delle caratteristiche di detti sistemi.

In questo esercizio si utilizza il modulo attivo come lettore di piu' moduli passivi (di piu' schede). Il lettore legge il codice di ogni scheda, lo visualizza sul monitor seriale e, se presente in una tabella interna, accende un led verde oppure un led rosso, a seconda del codice di azione a lui associato. Il programma puo' memorizzare i codici per un massimo di 10 schede e potrebbe ovviamente eseguire azioni diverse per ognuna di esse. Pur limitando le azioni alla semplice accensione di un led verde o di un led rosso, se si desidera replicare l'esercizio bisogna "specializzare" il programma con i codici delle schede che si intendono utilizzare, indicando anche, per ognuna di esse, il tipo di azione da intraprendere (accensione del led verde oppure accensione del led rosso).



Per fare questo si deve:

- Assemblare il circuito, compilare il programma e trasferirlo su Arduino;
- aprire il monitor seriale (ctrl + maiusc + M) oppure strumenti -> monitor seriale;
- strisciare le tessere che si intendono utilizzare;
- leggere sul monitor seriale il codice identificativo di ogni tessera (quattro numeri)
- inserire i quattro numeri di ogni scheda nella tabella codici, presente a partire dalla riga 82 del programma (attenzione a mantenere tutte

le virgole, limitarsi ad inserire i nuovi numeri al posto di quelli preesistenti)

- inserire, nella tabella dei codici di azione (primo elemento in riga 95 del programma) i codici di azione che si desiderano vengano intrapresi alla lettura della corrispondente scheda (1 accende il led verde, 2 accende il led rosso)
- ricompilare e provare il programma

Prima di procedere alla compilazione dello sketch e' anche necessario installare, se non gia' fatto, 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)

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 e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)

Arduino: RFID azioni diverse per ogni scheda – RFID: different action for each card

- [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



Some guidance on RFID principles are present in the example 36 ([here the sheet](#)) so now we don't talk about characteristics of this system.

In this example, we use the active module as reader for more RFID card. The reader reads code of each card, displays it on serial monitor and, if code is present in an internal table, lights a green light or a red light, depending on the associated action code.

The program can store codes for maximum ten cards and may of course perform different actions for each of them. While limiting the action to the simple lighting of a green or red light, if you want to replicate the example, you must "specialize" the program with the identification code of cards you are using, along with, for each, the action type to be performed (the green or red led).

To do this you must:

- Assemble the circuit, compile the program and transfer it on Arduino;
- open the serial monitor (ctrl + shift + M) or tools -> Serial monitor;
- swipe the cards that you intend to use;
- read on serial monitor the identification code of each card (four numbers)
- insert the four numbers of each card in the code table, starting on line 82 of program (careful to keep all commas, limit yourself to enter new numbers in place of existing ones)
- include in the table of action codes (first element in line 95 of program) the action codes which are to be undertaken at reading the corresponding card (1 green led, 2 red led)
- recompile and test the program

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

- MFRC522.h found [here](#)

For library installation, see process shown in previous exercises, and summarized in:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation need verification: the library 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.

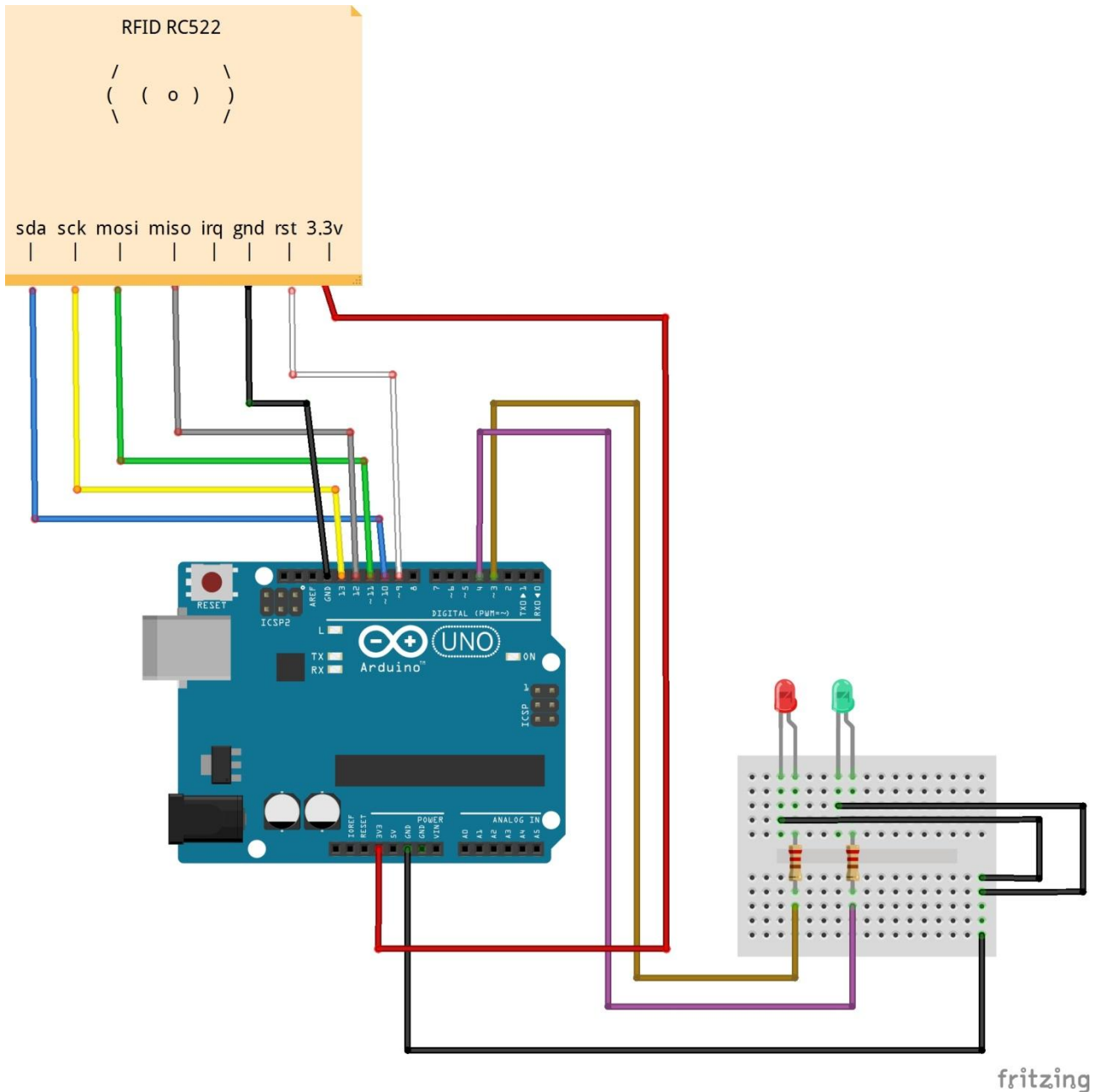
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

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)
- due resistenze da 220 ohm

Schema



Programma

```
/* 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. Con questo programma e' possibile memorizzare i
 * codici di max 10 schede di tipo mifare RC522 ed eseguire azioni diverse per ogni scheda.
 * Il programma riconosce le schede valide (e cioe' le schede i cui codici sono memorizzati)
```

Arduino: RFID azioni diverse per ogni scheda – RFID: different action for each card

```
* e per ogni scheda lancia l'esecuzione di azioni diverse, a seconda del "codice di azione"
* associato ad ogni scheda.
*
* il cuore del programma e' rappresentato dalle due tabelle "tabellacodici" e tabellazioni".
* Nella prima sono memorizzati i codici delle schede che devono essere riconosciute mentre
* nella seconda sono memorizzati i codici di azione relativi ad ogni singola scheda.
*
* in questo esercizio ci limitiamo ad accendere un led verde oppure un led rosso a seconda
* del codice azione rilevato. Nulla osta, ovviamente, che possano essere codificate azioni
* diverse, specifiche per ogni scheda.
*
* Alcune parti di questo programma sono derivate 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
*
* Attention: cut and paste from PDF to IDE loses formatting. to restore it press CTRL + T
* With this program you can store codes of up to 10 RC522mifare type cards and perform different
* actions for each card.
* the heart of the program are the two tables "tabellacodici" and "tabellazioni". "In the first are
* stored the codes of the cards which must be recognized while in the second are stored action
* codes relating to each single card.
* In this exercise we simply turn on a green or a red light depending on the action code detected.
* Obviously may be encoded different actions, specific for each card.
* Note: The MFRC522 library uses some predefined Arduino pins, whereby, to be able to repeat this
* exercise, it's necessary to respect the following connections between pins of the RC522 board and
* Arduino:
* -----
* .....pin .....Arduino .....Arduino ...Arduino.....Arduino..... Arduino
* .....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
*/

#include <SPI.h> // richiama la libreria di gestione del bus SPI
#include <MFRC522.h> // richiama la libreria di gestione del lettore rc522
#define RST_PIN 9
#define SS_PIN 10
#define ledrosso 3
#define ledverde 4
//
MFRC522 mfrc522(SS_PIN, RST_PIN); // Crea un "oggetto" MFRC522
//
char codice [5]; // variabile di memorizzazione del codice identificativo
// presente sulla scheda intercettata dal lettore
//
// Le prossime due tabelle (tabellacodici e tabellazioni) sono tabelle parallele, utilizzano cioe'
// il medesimo indice. La prima e' utilizzata per verificare la validita' o meno di una scheda
// mentre la seconda, fornisce, a parita' di indice di tabellacodici, il codice delle azioni da
// intraprendere. Se, ad esempio, i codici della scheda letta corrispondono ai codici del terzo
// elemento in tabellacodici, l'azione da intraprendere e' quella indicata dal codice presente nel
// terzo elemento della tabellazioni.
//
// the next two tables are "twin tables". This means that use the same index. In tabellacodici you
// must store code of each card (four issues) while in codiciazione table must be stored the
// associated action codes
//
char tabellacodici [41] = // tabella dei codici delle schede (valori ascii)
{
  0, 0, 0, 0, // <-- inserire i codici della prima scheda - first card code
  0, 0, 0, 0, // <-- inserire i codici della seconda scheda - second card code
```

Arduino: RFID azioni diverse per ogni scheda – RFID: different action for each card

```
0, 0, 0, 0, // <-- inserire i codici della terza scheda - third card code
0, 0, 0, 0, // <-- inserire i codici della quarta scheda and so on
0, 0, 0, 0, // <-- inserire i codici della quinta scheda
0, 0, 0, 0, // <-- inserire i codici della sesta scheda
0, 0, 0, 0, // <-- inserire i codici della settima scheda
0, 0, 0, 0, // <-- inserire i codici della ottava scheda
0, 0, 0, 0, // <-- inserire i codici della nona scheda
0, 0, 0, 0, // <-- inserire i codici della decima scheda
};
char tabellazioni [11] =
{
  1, // <-- inserire il codice di azione per la prima scheda in tab.codici first card action code
  2, // <-- inserire il codice di azione per la seconda scheda in tab.codici second card action code
  0, // <-- inserire il codice di azione per la terza scheda in tab.codici third card action code
  0, // <-- inserire il codice di azione per la quarta scheda in tab.codici and so on
  0, // <-- inserire il codice di azione per la quinta scheda in tabellacodici
  0, // <-- inserire il codice di azione per la sesta scheda in tabellacodici
  0, // <-- inserire il codice di azione per la settima scheda in tabellacodici
  0, // <-- inserire il codice di azione per la ottava scheda in tabellacodici
  0, // <-- inserire il codice di azione per la nona scheda in tabellacodici
  0, // <-- inserire il codice di azione per la decima scheda in tabellacodici
};
int i = 0; // indice utilizzato per scorrere la tabella di memorizzazione dei caratteri
// provenienti dal lettore
int semaforo = 0; // semaforo utilizzato per segnalare l'uguaglianza tra il codice letto ed
// un codice in tabellacodici: 0 = codici non valido; 1 = codice valido
int codiceazione = 0; // codice dell'azione da intraprendere per la scheda appena letta
int indicetab = 0; // indice di scorrimento della tabellacodici
//
//
/* ***** routine esecuzione attivita' per codice azione = 1 *****/
// ***** action code 1 routine *****/
void azione1(void)
{
  digitalWrite (ledverde, HIGH); // accende il led verde
  digitalWrite (ledrosso, LOW); // spegne il led rosso, qualora fosse acceso
}
//
//
/* ***** routine esecuzione attivita' per codice azione = 2 *****/
// ***** action code 2 routine *****/
void azione2 (void)
{
  digitalWrite (ledrosso, HIGH); // accende il led rosso
  digitalWrite (ledverde, LOW); // spegne il led verde, qualora fosse acceso
}
//
void setup()
{
  Serial.begin(9600); // Inizializza la comunicazione seriale con il pc
  SPI.begin(); // Inizializza il bus SPI
  mfrc522.PCD_Init(); // Inizializza il lettore MFRC522
  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
}
//
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'
  // del tipo MIFARE
  }
  Serial.print ("letta scheda con codice identificativo: ");
  for (i = 0; i < mfrc522.uid.size; i++) // loop di acquisizione dei codici 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 della
  // scheda
  }
  Serial.println (" ");
}
```

Arduino: RFID azioni diverse per ogni scheda – RFID: different action for each card

```
//
// ***** verifica la validita' della scheda letta e, se valida, rileva il codice di azione *****
// ***** check validity of readed card and, if valid, find the action code *****
codiceazione = 99; // inserisce inizialmente un valore non valido in codiceazione
for (indicetab = 0; indicetab < 10; indicetab++) // scorre la tabella dei codici validi
{
  semaforo = 1; // pone preventivamente a 1 un semaforo
  for (i = 0; i < 4; i++) // verifica, carattere per carattere, se il codice letto
// // corrisponde al codice valido memorizzato nell'elemento in
// // esame della tabellacodici
  {
    if (! (codice [i] == tabellacodici [indicetab * 4 + i])) // se il carattere letto non
// // corrisponde al carattere in tabellacodici
      semaforo = 0; // spegne il semaforo (la scheda letta non corrisponde alla
// // scheda in tabellacodici)
  }
  if (semaforo == 1) // se il semaforo vale 1 (se la scheda letta e' presente in
// // taballecodici)
  {
    codiceazione = tabellazioni [indicetab]; // memorizza il codice di azione corrispondente alla
// // scheda letta
  }
// //
// // verifica il codice di azione associato alla scheda appena letta ed esegue routine diverse a
// // seconda del suo valore
// // *****check action code and launch different action*****

  if (codiceazione == 1)
    azione1(); // lancia la routine delle attivita' da svolgere con codice azione = 1
  if (codiceazione == 2)
    azione2(); // lancia la routine delle attivita' da svolgere con codice azione = 2
  if (codiceazione == 99)
    Serial.println ("scheda non presente in tabella"); // card out of table
  delay (2000); // attende due secondi prima di verificare la presenza di una nuova scheda
}
```