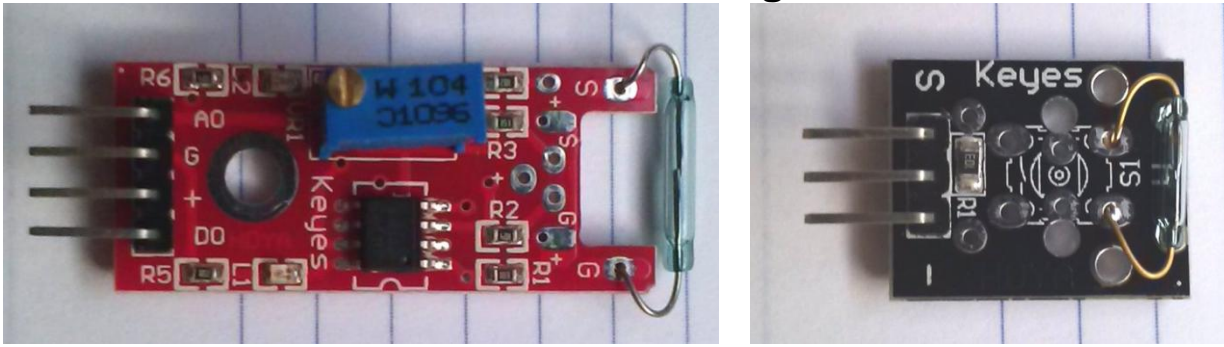


D – sistema d'allarme con sensori magnetici



Il sistema di allarme proposto in questa nota prevede l'utilizzo di sensori reed per rilevare l'apertura di porte e/o finestre. Il programma prevede due sensori, ma il loro numero puo' essere facilmente ampliato. Il sistema, oltre ai sensori ed ai relativi magneti, utilizza anche un display 1620 a due porte ([vedi esercizio 18bis](#)), un buzzer, due led ed un pulsante. Nella sua forma essenziale, puo' comunque funzionare anche con un solo sensore (non necessariamente magnetico e non necessariamente reed), un led, un buzzer ed un pulsante. Sara' in questo caso sufficiente disabilitare le istruzioni che riferiscono il led rosso, il secondo sensore ed il display lcd. Sara' inoltre necessario sostituire gli avvisi di "situazione anomala" con un lampeggio del led.

Il sistema, che si attiva nel momento in cui un magnete si allontana dal suo sensore, e' utilizzabile per allarmare porte e/o finestre fissando il magnete alla parte mobile dell'infisso ed il sensore allo stipite.

Come in ogni sistema di allarme e' previsto un ritardo di qualche secondo tra il momento dell'avvio ed il momento dell'entrata in servizio (per consentire al padrone di casa di uscire e chiudere la porta allarmata) e di qualche altro secondo tra il momento dell'apertura della porta e l'attivazione del buzzer (per consentirne la disattivazione al rientro del padrone di casa). Il buzzer ed il led, una volta attivati, non si interrompono nemmeno se si richiude la porta o la finestra, ma continuano a suonare ed a lampeggiare per il tempo programmato o fino a quando non si preme il pulsante di cessato allarme (il pulsante "reset" di Arduino).

Il pulsante inserito nello schema funge da attivatore del sistema: quando lo si preme il sistema controlla la situazione dei sensori (devono essere tutti chiusi) ed inizia il countdown (visualizzato anche sul display) al termine del quale si arma ed accende il led giallo. In caso di allarme (apertura di uno dei due sensori reed) parte un nuovo countdown, sempre visualizzato sul display, al termine del quale il buzzer inizia a suonare ed il led rosso a lampeggiare. Per spegnere l'allarme prima del tempo previsto a programma o per disarmare il sistema bisogna premere il tasto reset di Arduino.

Al posto del buzzer e del led rosso possono ovviamente essere attivati dei rele' che a loro volta attivano una vera sirena ed un vero segnale luminoso, cosi' come e' possibile associare anche una batteria tampone che consenta al sistema di operare anche in assenza di energia da rete.

Nota: Questo prototipo 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.

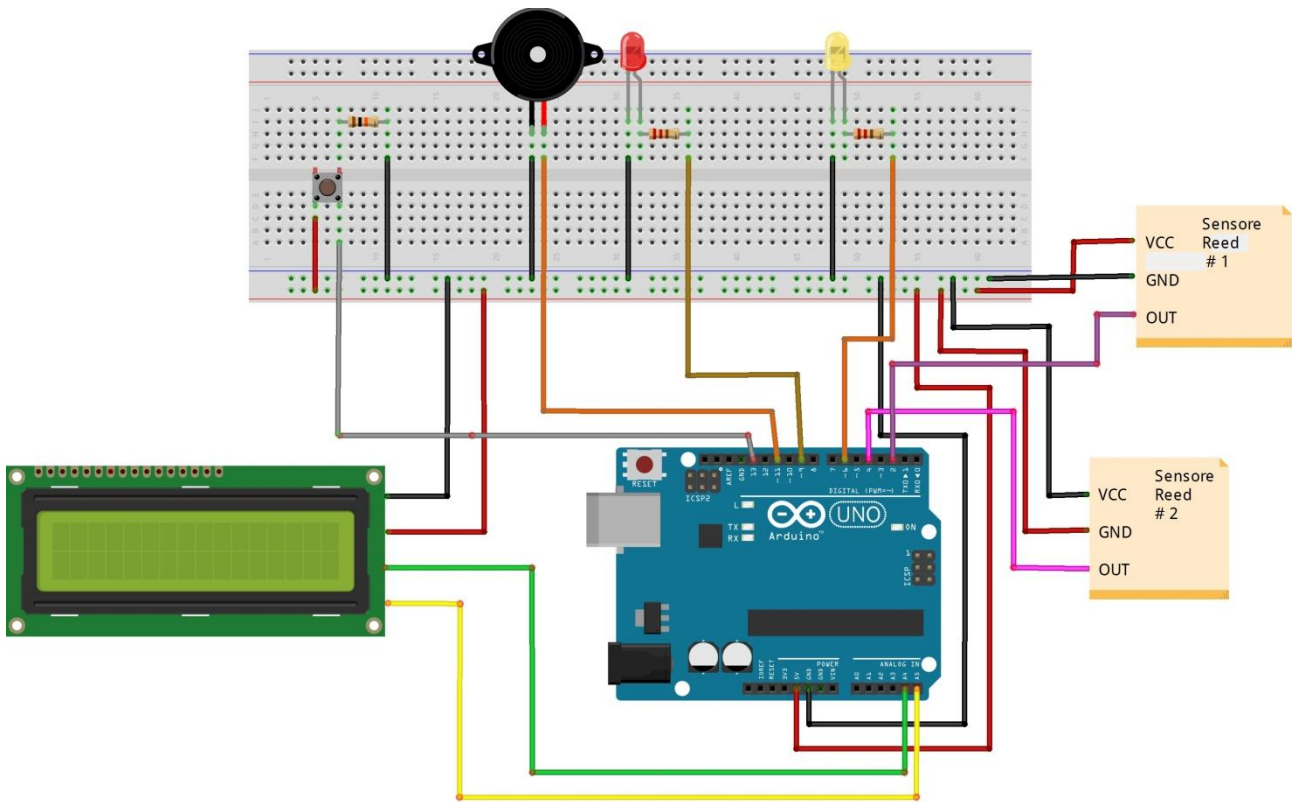
- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

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

Materiali

- due sensori reed montati su basetta
- due magneti
- un display 1602 a due porte ([vedi esercizio 18bis](#))
- un buzzer
- due led (uno rosso ed uno giallo)
- tre resistenze: due da 220 ohm ed una da 10 k ohm
- un pulsante

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. Nel trasferimento si perdono anche gli spazi compresi tra
apici (" "), per cui una volta trasferito il programma nell'IDE bisogna intervenire manualmente
per reintrodurre gli spazi perduti (vedi note in rosso nella versione pdf del programma).
*
```

```
Questo programma configura un sistema di allarme con due sensori
```

```
*
```

```
l'impianto utilizza due sensori magnetici, un display lcd a due porte, due led,
un buzzer, un pulsante, due resistenze da 220 ohm ed una resistenza da 10k ohm.
Lo schema e le spiegazioni sono reperibili nel menu "prototipi" di questo sito:
```

```
*
```

```
http://giocarduino.altervista.org/
```

```
*
```

```
Il sistema si arma premendo il pulsante e si disarma premendo il tasto reset di Arduino
Per dare tempo al padrone di casa di uscire dopo averlo armato e di disarmarlo al
momento del rientro, il sistema prevede un ritardo di una decina di secondi
nell'attivazione ed un altro analogo ritardo nel lancio dell'allarme
```

Arduino: sistema d'allarme con sensori magnetici

Sul display appaiono informazioni sullo stato, su eventuali problemi, sul countdown al momento dell'attivazione del sistema e nel lancio dell'allarme

```
*/

#include <Wire.h> // libreria wire presente, di fault, nell'IDE
#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd a due porte(vedi esercizio
// 18 bis nel sito sopra indicato)
// nota: i terminali SDA e SCL del display devono essere collegati, rispettivamente, alle porte
// analogiche 4 e 5
//-----addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // definisce la tipologia del display
//
// *****definizione delle costanti *****
//
const int sensore1 = 2; // il sensore numero 1 e' collegato alla porta 2
const int sensore2 = 4; // il sensore numero 2 e' collegato alla porta 4
const int ledgiallo = 6; // il led giallo e' collegato alla porta 6
const int ledrosso = 9; // il led rosso e' collegato alla porta 9
const int buzzer = 11; // il buzzer e' collegato alla porta 11
const int pulsante = 13; // il pulsante e' collegato alla porta 13
const int duratallarme = 50; // tempo in secondi di durata allarme (sirena e led lampeggiante)
//
// *****definizione delle variabili *****
//
int statosensore1 = 0; // variabile in cui viene acquisito lo stato del sensore 1
int statosensore2 = 0; // variabile in cui viene acquisito lo stato del sensore 2
int sensoriok = 0; // contiene 1 se entrambi i sensori sono chiusi
int sistemarmato = 0; // variabile in cui viene inserito 1 quando il sistema e' armato
int situazionepulsante = 0; // variabile in cui viene acquisito lo stato del pulsante
int tempocountdown = 0; // durata del countdown
int secondiresidui = 0; // secondi mancanti alla fine del countdown
int secondipassati = 0; // secondi trascorsi da inizio countdown
int i = 0; // variabile utilizzata nei cicli di for
//
//***** routine di pulizia del display *****
//
void puliscischermo (void)
{
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print(" "); // attenzione: inserire 16 spazi tra i due apici
  lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
  lcd.print(" "); // attenzione: inserire 16 spazi tra i due apici
}

//***** routine di verifica dei sensori *****
void verificasensori (void)
{
  sensoriok = 0;
  statosensore1 = digitalRead(sensore1); // acquisisce lo stato del sensore 1
  statosensore2 = digitalRead(sensore2); // acquisisce lo stato del sensore 2
  if (statosensore1 == HIGH & statosensore2 == HIGH) // se entrambi i sensori sono chiusi
    sensoriok=1; // inserisce il valore 1 in sensoriok
}
//
//***** routine di visualizzazione dello stato dei sensori *****
//
void displaysituazione sensori (void)
{
  puliscischermo ();
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print("situaz. sensori");
  lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
  lcd.print("s1 = ");
  if (statosensore1 == 1)
    lcd.print ("ok");
  else
    lcd.print ("ko");
  lcd.print(" s2 = ");
  if (statosensore2 == 1)
    lcd.print ("ok");
  else
    lcd.print ("ko");
  delay (3000);
}
//
// ***** routine di countdown *****
//
```

Arduino: sistema d'allarme con sensori magnetici

```
void countdown (void)
{
  for (secondipassati = 0; secondipassati <= tempocountdown; secondipassati++)
  {
    lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
    secondiresidui = tempocountdown - secondipassati;
    lcd.print(secondiresidui);
    lcd.print (" secondi ");
    delay (1000); // attende un secondo prima di riprendere il ciclo di for
  }
}
//
// ***** routine di attivazione del sistema *****
//
void armailsistema (void)
{
  verificasensori(); // verifica lo stato iniziale dei sensori
  while (sensoriok == 0) // se la situazione non e' corretta (uno o due sensori sono aperti)
  {
    displaysituazione sensori ();
    puliscischermo ();
    lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
    lcd.print("situaz. anomala");
    lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
    lcd.print("verif. sensore ");
    if (statosensore1 == 0)
      lcd.print ("1");
    if (statosensore2 == 0)
      lcd.print ("2");
    delay (3000); // attende tre secondi prima di verificare nuovamente la situazione dei sensori
    verificasensori (); // verifica i sensori e riprende il test di while
  }

  // usciti dal ciclo di while, la situazione e' corretta e si procede con l'attivazione
  displaysituazione sensori ();
  puliscischermo ();
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print("sensori ok");
  for (i=0; i<=3; i++) // segnala, lampeggiando tre volte, che il sistema e' attivabile
  {
    lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
    lcd.print("sist. attivabile");
    delay (500); // attende mezzo secondo
    lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
    lcd.print(" "); // attenzione, inserire 16 spazi tra i due apici
    delay (500); // attende mezzo secondo e riprende il ciclo di for
  }

  puliscischermo ();
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print("attivazione tra ");
  tempocountdown = 10; // secondi di delay da far trascorrere prima di armare il sistema
  countdown (); // lancia la routine di countdown
  digitalWrite (ledgiallo, HIGH); // accende la spia luminosa (led giallo) che segnala che
  // il sistema e' armato
  puliscischermo ();
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print("sistema");
  lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga (carattere 0 e riga 1)
  lcd.print("attivato");
  sistemarmato = 1; //inserisce 1 nel semaforo di sistema armato
}

// ***** routine di allarme *****
void lanciaallarme (void)
{
  displaysituazione sensori (); // Visualizza la situazione dei sensori
  puliscischermo ();
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print("allarme tra ");
  tempocountdown = 10; // secondi di delay da far trascorrere prima di far scattare l'allarme
  countdown (); //lancia la routine di countdown
  puliscischermo ();
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga
  lcd.print ("allarme ");
  lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
  lcd.print("verif. sensore ");
}
```

Arduino: sistema d'allarme con sensori magnetici

```
if (stato sensore1 == 0)
  lcd.print ("1");
if (stato sensore2 == 0)
  lcd.print ("2");
for (i=1; i<= duratallarme; i++) // ciclo di allarme; per allungare o accorciare
//                               il tempo di allarme, variare il valore di duratallarme
{
  digitalWrite (buzzer, HIGH);
  digitalWrite (ledrosso, LOW);
  delay (500);
  digitalWrite (buzzer, LOW);
  digitalWrite (ledrosso, HIGH);
  delay (500);
  digitalWrite (ledrosso, LOW);
}
}

void setup()
{
  lcd.begin(16,2); // inizializza il display (16 caratteri per due righe)
  lcd.backlight(); // illumina lo sfondo del display
  puliscischermo ();
  pinMode (sensore1, INPUT); // definisce il sensore 1 come unita' di input
  pinMode (sensore2, INPUT); // definisce il sensore 2 come unita' di input
  pinMode (pulsante, INPUT); // definisce il pulsante come unita' di input
  pinMode (ledgiallo, OUTPUT); // definisce il led giallo come unita' di output
  pinMode (ledrosso, OUTPUT); // definisce il led rosso come unita' di output
  pinMode (buzzer, OUTPUT); // definisce il buzzer come unita' di output
}

void loop ()
{
  if (sistemarmato == 0) // se il sistema non e' ancora armato
  {
    situazionepulsante = digitalRead(pulsante); // acquisisce lo stato del pulsante
    if (situazionepulsante == HIGH) // se il pulsante e' premuto
      armailsistema (); // lancia la routine di attivazione
  }

  if (sistemarmato == 1) // se il sistema e' armato
  {
    verificasensori (); //verifica la situazione dei sensori
    if (sensoriok == 0) // se i sensori non sono chiusi
      lanciallarme (); //lancia la routine di allarme
  }
}
```