**Arduino lezione 05 controllo presenza**

Pubblicato il [marzo 21, 2011](http://www.maffucci.it/2011/03/21/arduino-lezione-05-controllo-presenza/) da [admin](http://www.maffucci.it/author/admin/%22%20%5Co%20%22Visualizza%20tutti%20gli%20articoli%20di%20admin)

Questa lezione nasce da un [commento di Aldo Biscotti](http://www.maffucci.it/2010/12/06/arduino-%E2%80%93-lezione-03-controlliamo-un-led-con-un-pulsante/comment-page-1/#comment-1066) *(seguite il link)*, che ha la necessità di realizzare l’illuminazione temporizzata delle scale di casa. Si sta cimentando in questo progetto e la sua richiesta di aiuto mi da parecchi spunti per la realizzazione di una lezione ad hoc per i miei allievi.
Per la realizzazione del suo progetto avrà la necessità di usare dei sensori di presenza, che in questa lezione saranno sostituiti da semplici pulsanti.

Dal commento che mi è stato lasciato desumo che la necessità descritta da Aldo è quella di comandare due lampade di due ambienti diversi mediante due distinti rilevatori di presenza.
Nel realizzare un primo prototipo di studio Aldo giustamente, visto che sta imparando a programmare Arduino, utilizza i pulsanti in sostituzione dei rilevatori e i diodi led in sostituzione delle lampade.

Questa la sua necessità:

*Pulsante 1:
(salita) start,
accensione del led 1,
dopo 2 secondi accensione led 2,
dopo 2 secondi spegnimento led 1,
dopo 2 secondi spegnimento led 2,
end.*

*Pulsante 2:
(discesa) start,
accensione del led 2,
dopo 2 secondi accensione led 1,
dopo 2 secondi spegnimento led 2,
dopo 2 secondi spegnimento led 1,
end.*

***Grazie Aldo.***

Risponderò ad Aldo prendendo in analisi il suo primo sketch e da questo ne svilupperò una serie di programmi in cui introdurrò nuovi concetti.

Innanzitutto realizziamo il primo circuito di test, costituito da due diodi led, una resistenza ed un pulsante, in questa lezione vedrete che utilizzerò sia una scheda Arduino UNO, che Arduino 2009, assolutamente simili dal punto di vista della programmazione:



Questo invece il disegno realizzato con [Fritzing](http://www.maffucci.it/2010/11/18/come-progettare-e-documentare-i-lavori-da-realizzare-con-arduino/)



Analizziamo il primo sketch.

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/21/arduino-lezione-05-controllo-presenza/)

1. // Esempio 1: accensione led
2.
3. **int** led\_1 = 8;
4. **int** led\_2 = 9;
5. **int** btn\_pin = 2;
6.
7. **void** setup() {
8. pinMode(led\_1, OUTPUT);
9. pinMode(led\_2, OUTPUT);
10. pinMode(btn\_pin, INPUT);
11. }
12.
13. **void** loop()
14. {
15. **int** in = digitalRead(btn\_pin);
16. **if** (in == LOW)
17. {
18. digitalWrite(led\_1, LOW);
19. delay (2000);
20. digitalWrite(led\_2, LOW);
21. delay (2000);
22. }
23. **else**
24. {
25. digitalWrite(led\_1, HIGH);
26. delay (2000);
27. digitalWrite(led\_2, HIGH);
28. delay (2000);
29. }
30. }

Non funziona molto bene, come si desume dal filmato vi sono alcune correzioni da fare:

Nell’if viene controllato se il pulsante non è stato premuto “in == LOW”, se vero vengono spenti in modo sequenziale i due LED, domanda:

Se una luce è già spenta perché aggiungere un ritardo?

Ciò implica che se il programma è in esecuzione nella prima parte dell’if non verrà rilevata la presenza di una persona al massimo per 4 secondi ed è come dire che appena entro nella stanza sarò al buio per 4 secondi prima che le luci vengano accese e quindi una persona potrà:

* fermarsi ed attendere 4 secondi
* oppure camminare per 4 secondi al buio compiendo circa 5 metri di spazio camminando, presumibilmente supererà la prima stanza al buio.

Questa la mia prima variazione:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/21/arduino-lezione-05-controllo-presenza/)

1. // Esempio 2: accensione temporizzata di un led in una sola direzione
2.
3. #define led\_1 8    // il pin 8 è usato per il LED
4. #define led\_2 9    // il pin 9 è usato per il LED
5. #define btn\_pin 2  // il pin 2 è usato per il PULSANTE 1
6.
7. // Variabili
8. **int** in = 0;
9.
10. **void** setup() {
11. pinMode(led\_1, OUTPUT);    // impostiamo il pin led\_1 come output
12. pinMode(led\_2, OUTPUT);    // impostiamo il pin led\_2 come output
13. pinMode(btn\_pin, INPUT);   // impostiamo il pin btn\_pin come input
14. }
15.
16. **void** loop()
17. {
18. in = digitalRead(btn\_pin); // assegna lo stato del pulsante
19. **if** (in == HIGH)            // controlla se il pulsante è stato premuto
20. {
21. digitalWrite(led\_1, HIGH);  // accendiamo il LED 1
22. delay (2000);               // attesa di 2 sec
23. digitalWrite(led\_2, HIGH);  // accendiamo il LED 2
24. delay (2000);               // attesa di 2 sec
25. digitalWrite(led\_1, LOW);   // spegniamo il LED 1
26. delay (2000);               // attesa di 2 sec
27. digitalWrite(led\_2, LOW);   // spegniamo il LED 2
28. }
29. }

Controllo se il pulsante e premuto, se vero inizia la sequenza di accensione, altrimenti se il pulsante non è premuto l’if non viene eseguito, le luci continuano a rimanere spente e ricomincia nuovamente il loop().
Ovviamente se durante la sequenza di accensione e spegnimento di 6 secondi viene premuto nuovamente il pulsante non accade nulla, vedremo più avanti che questa soluzione potrebbe causare qualche problema.

Vediamo adesso come utilizzare due pulsanti (o rilevatori di presenza) per governare l’accensione in senso opposto, il circuito realizzato è il seguente:





Questo lo sketch

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/21/arduino-lezione-05-controllo-presenza/)

1. // Esempio 3: accensione temporizzata di due led
2. // in due direzioni con uso dell'if
3.
4. #define led\_1 8    // il pin 8 è usato per il LED
5. #define led\_2 9    // il pin 9 è usato per il LED
6. #define btn\_pin1 2  // il pin 2 è usato per il PULSANTE 1
7. #define btn\_pin2 3  // il pin 3 è usato per il PULSANTE 2
8.
9. // Variabili
10. **int** in1 = 0;        // variabile per memorizzare lo stato del PULSANTE 1
11. **int** in2 = 0;        // variabile per memorizzare lo stato del PULSANTE 2
12.
13. **void** setup() {
14. pinMode(led\_1, OUTPUT);
15. pinMode(led\_2, OUTPUT);
16. pinMode(btn\_pin1, INPUT);
17. pinMode(btn\_pin2, INPUT);
18. }
19.
20. **void** loop()
21. {
22. in1 = digitalRead(btn\_pin1);
23. in2 = digitalRead(btn\_pin2);
24.
25. **if** (in1 == HIGH && in2 == LOW)  // se PULSANTE 1 premuto e PULSANTE 2 no sequenza 1
26. {
27. digitalWrite(led\_1, HIGH);
28. delay (2000);
29. digitalWrite(led\_2, HIGH);
30. delay (2000);
31. digitalWrite(led\_1, LOW);
32. delay (2000);
33. digitalWrite(led\_2, LOW);
34. }
35. **if** (in1 == LOW && in2 == HIGH)  // se PULSANTE 1 premuto e PULSANTE 2 no sequenza 2
36. {
37. digitalWrite(led\_2, HIGH);
38. delay (2000);
39. digitalWrite(led\_1, HIGH);
40. delay (2000);
41. digitalWrite(led\_2, LOW);
42. delay (2000);
43. digitalWrite(led\_1, LOW);
44. }
45. }

Come potete notare ho usato l’operatore logico AND (&&) che effettua un controllo su quale dei pulsanti è acceso in questo modo controllo la direzione di accensione.

Per chiarire meglio ho realizzato uno schema che visualizza meglio cosa accade:



Si ricordi che la progettazione viene fatta usando pulsanti, ma voi estrapolate e pensate a dei rilevatori di presenza.

Supponiamo che la direzione sia da destra a sinistra, la persona 1 viene rilevata dal sensore R1, si accende la luce L1 per 2 secondi, entra nella stanza 2 e supponendo che non sia più veloce di due secondi, rischiando di entrare in una stanza buia, viene rilevato dal sensore R2, si accende la luce L2 e dopo 4 secondi si spegne la luce L1.

Ma cosa accade se nei primi 2 secondi entra un’altra persona nella prima stanza a destra?
La seconda persona resta al buio.
Stesso inconveniente accade nella direzione opposta da sinistra verso destra.

Quindi per superare questo incoveniente è necessario che l’accensione delle luci non duri solamente per un tempo prefissato, ma per tutto il tempo in cui la persona è presente nel locale, dopo di che la luce può spegnersi.

La soluzione più semplice di tutte è quella descritta dallo sketch che segue, si ricordi che premere il pulsante corrisponde a: “persona rilevata”

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/21/arduino-lezione-05-controllo-presenza/)

1. // Esempio 4: accensione di due led comandati da due pulsanti
2.
3. #define led\_1 8    // il pin 8 è usato per il LED
4. #define led\_2 9    // il pin 9 è usato per il LED
5. #define btn\_pin1 2  // il pin 2 è usato per il PULSANTE 1
6. #define btn\_pin2 3  // il pin 3 è usato per il PULSANTE 2
7.
8. // Variabili
9. **int** in1 = 0;
10. **int** in2 = 0;
11.
12. **void** setup() {
13. pinMode(led\_1, OUTPUT);
14. pinMode(led\_2, OUTPUT);
15. pinMode(btn\_pin1, INPUT);
16. pinMode(btn\_pin2, INPUT);
17. }
18.
19. **void** loop()
20. {
21. in1 = digitalRead(btn\_pin1);
22. in2 = digitalRead(btn\_pin2);
23.
24. **if** (in1 == HIGH)                  // PULSANTE 1 premuto
25. {
26. digitalWrite(led\_1, HIGH);      // accensione LED 1
27. }
28. **else**
29. {
30. digitalWrite(led\_1, LOW);      // seil PULSANTE 1 non premuto LED 1 spento
31. }
32.
33. **if** (in2 == HIGH)                  // PULSANTE 2 premuto
34. {
35. digitalWrite(led\_2, HIGH);      // accensione LED 2
36. }
37. **else**
38. {
39. digitalWrite(led\_2, LOW);      // seil PULSANTE 2 non premuto LED 2 spento
40. }
41.
42. }

Per evitare che ci sia uno spegnimento brusco dell’illuminazione appena si esce dalla stanza, ritardo lo spegnimento dell’illuminazione della stanza che ho appena abbandonato:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/21/arduino-lezione-05-controllo-presenza/)

1. // Esempio 5: accensione di due led comandati da due pulsanti
2. // con ritardo di spegnimento di due secondi
3.
4. #define led\_1 8    // il pin 8 è usato per il LED
5. #define led\_2 9    // il pin 9 è usato per il LED
6. #define btn\_pin1 2  // il pin 2 è usato per il PULSANTE 1
7. #define btn\_pin2 3  // il pin 3 è usato per il PULSANTE 2
8.
9. // Variabili
10. **int** in1 = 0;
11. **int** in2 = 0;
12.
13. **void** setup() {
14. pinMode(led\_1, OUTPUT);
15. pinMode(led\_2, OUTPUT);
16. pinMode(btn\_pin1, INPUT);
17. pinMode(btn\_pin2, INPUT);
18. }
19.
20. **void** loop()
21. {
22. in1 = digitalRead(btn\_pin1);
23. in2 = digitalRead(btn\_pin2);
24.
25. **if** (in1 == HIGH){                 // PULSANTE 1 premuto
26. digitalWrite(led\_1, HIGH);      // accensione LED 1
27. delay (2000);                   // ritardo di 2 sec
28. }
29. **else**
30. {
31. digitalWrite(led\_1, LOW);       // se il PULSANTE 1 non premuto LED 1 spento
32. }
33.
34. **if** (in2 == HIGH){                 // PULSANTE 2 premuto
35. digitalWrite(led\_2, HIGH);      // accensione LED 2
36. delay (2000);                   // ritardo di 2 sec
37. }
38. **else**
39. {
40. digitalWrite(led\_2, LOW);        // seil PULSANTE 2 non premuto LED 2 spento
41. }
42. }

Nella prossima lezione al fine di rendere più utile e gradevole il controllo sull’illuminazione della stanza vedremo come aumentare e diminuire gradualmente l’illuminazione usando una tecnica che sfrutta la **modulazione di larghezza di impulso (PWM)**

**Aldo Biscotti** *scrive:*

[marzo 19, 2011 alle 12:07 pm](http://www.maffucci.it/2010/12/06/arduino-lezione-03-controlliamo-un-led-con-un-pulsante/comment-page-1/#comment-1066)

Buon giorno Professore e a tutti i lettori,
mi chiamo Aldo Biscotti e sono un esordiente in questa materia.
Sono affascinato dalle tue lezioni, dove sto anzi spero di imparare un po’ di questa tecnica
di programmazione.
Sto cercando di cimentarmi con arduino 2009, vorrei realizzare un software che mi permetta di illuminare le scale di casa, non ti nascondo che forse, anzi sicuramente, mi sono avventurato in un
progetto troppo ambizioso x me, questa cosa ormai e’ diventata un’ ossessione prendendomi tutto il tempo libero e buona parte della notte.
Prima di iniziare, ti voglio ringraziare per la pubblicazione della lezione 3: come accendere un led con un interruttore, e seguendo i tuoi consigli, ho continuato a provare, provare e ancora provare e con grande soddisfazione mi ha fatto raggiungere parte dell’ obbiettivo, ovvero l’inserimento di un interruttore per far partire un software non ciclato.
Avevo gia’ letto la lezione, ma non le ho dato la giusta importanza proprio perche’ sono un principiante e ho iniziato dall’ effetto che desideravo, riflettendoci ho capito che avrei iniziato dalla fine. Avendo gia’ una parte di programma invece di adattare il sensore o l’interruttore a quest’ultimo, ho pensato fosse piu’ logico prendere lo sketch del pulsante, modificarlo adattandolo alle mia necessita’ e con mia grossa sorpresa ha funzionato.
Ora sono arrivato al momento del raddoppio, cioe’ l’ inserimento di un secondo interruttore che serve ad avviare in modo inverso l’ accensione dei led, guarda ho provato in tutti i modi ma niente da fare, non funziona. Se hai tempo, potresti dare un’ occhiata ai miei sketch e come ho interpretato il software, e darmi un consiglio?
Ecco cosa vorrei realizzare: l’ esempio riportato e’ di due led che simulano due gradini, ovviamente basta aggiungere tanti led quanti gradini bisogna illuminare e i tempi di delay sono simbolici.
TARGET: ( I pulsanti 1 e 2 saranno sostituiti da sensori di presenza anche se non so cosa usare l’importante e’che non facciano partire il ciclo al passaggio del cane o del gatto. )
Pulsante 1: (salita) start, accensione del led 1, dopo 2 secondi accensione led 2, dopo 2 secondi spegnimento led 1, dopo 2 secondi spegnimento led 2, end.
Pulsante 2: (discesa) start, accensione del led 2, dopo 2 secondi accensione led 1, dopo 2 secondi spegnimento led 2, dopo 2 secondi spegnimento led 1, end.

Inizio con la prima modifica dello sketch dell’ interruttore:

int led\_1 = 8;

int led\_2 = 9;

int btn\_pin = 2;

void setup() {

 pinMode(led\_1, OUTPUT);

 pinMode(led\_2, OUTPUT);

 pinMode(btn\_pin, INPUT);

}

void loop()

{

 int in = digitalRead(btn\_pin);

 if (in == LOW)

 {

 digitalWrite(led\_1, LOW);

 delay (2000);

 digitalWrite(led\_2, LOW);

 delay (2000);

 }

 else

 {

 digitalWrite(led\_1, HIGH);

 delay (2000);

 digitalWrite(led\_2, HIGH);

 delay (2000);

 }

}

Questo sketch gira benissimo: start, accensione del led 1, dopo 2 secondi accensione led 2, dopo 2 secondi spegnimento led 1, dopo 2 secondi spegnimento led 2, end.
Proprio quello che volevo, non male vero? Pensare che solo due mesi fa pensavo ad Arduino come ad un vecchio compagno delle elementari.
Dovendo inserire il secondo pulsante per il senso inverso, inizialmente ho pensato a questo sketch:

int led\_1 = 8;

int led\_2 = 9;

int btn\_pin\_1 = 2;

int btn\_pin\_2 = 3;

void setup() {

 pinMode(led\_1, OUTPUT);

 pinMode(led\_2, OUTPUT);

 pinMode(btn\_pin\_1, INPUT);

 pinMode(btn\_pin\_2, INPUT);

}

void loop()

{

 int in = digitalRead(btn\_pin\_1);

 if (in == LOW)

 int in = digitalRead(btn\_pin\_2);

 if (in == LOW)

 {

 digitalWrite(led\_1, LOW);

 delay (2000);

 digitalWrite(led\_2, LOW);

 delay (2000);

 }

 else

 {

 digitalWrite(led\_1, HIGH);

 delay (2000);

 digitalWrite(led\_2, HIGH);

 delay (2000);

 }

 {

 digitalWrite(led\_2, LOW);

 delay (2000);

 digitalWrite(led\_1, LOW);

 delay (2000);

 }

 {

 digitalWrite(led\_2, HIGH);

 delay (2000);

 digitalWrite(led\_1, HIGH);

 delay (2000);

 }

}

Ho eseguito il test prima del upload, perfetto non dava errori, ma scaricandolo su arduino una gran delusione, non rispetta i tempi, cicla, insomma un fallimento, ho abbandonato lo sketch.
Sono trooooppo scarso……….
Il giorno seguente a forza di pensarci mi e’ venuta una nuova idea:

int led\_1 = 8;

int led\_2 = 9;

int btn\_pin\_1 = 2;

int btn\_pin\_2 = 3;

void setup() {

 pinMode(led\_1, OUTPUT);

 pinMode(led\_2, OUTPUT);

 pinMode(btn\_pin\_1, INPUT);

 pinMode(btn\_pin\_2, INPUT);

}

void loop()

{

 int in = digitalRead(btn\_pin\_1);

 if (in == LOW)

 {

 digitalWrite(led\_1, LOW);

 delay (2000);

 digitalWrite(led\_2, LOW);

 delay (2000);

 }

 else

 {

 digitalWrite(led\_1, HIGH);

 delay (2000);

 digitalWrite(led\_2, HIGH);

 delay (2000);

 }

 int in = digitalRead(btn\_pin\_2);

 if (in == LOW)

 {

 digitalWrite(led\_2, LOW);

 delay (2000);

 digitalWrite(led\_1, LOW);

 delay (2000);

 }

 else

 {

 digitalWrite(led\_2, HIGH);

 delay (2000);

 digitalWrite(led\_1, HIGH);

 delay (2000);

 }

}

Mi sembrava piu’ equilibrato piu’ pulito del precedente ma non va’, mi da’ errore sul secondo pulsante.
Mi sorge un dubbio: si puo’ mettere 2 o piu’ interruttori per eseguire un software? Secondo me si. COME SI FA’?
Puoi darmi un aiuto?
Ti ringrazio anticipatamente un caloroso saluto .
Aldo

[Rispondi](http://www.maffucci.it/2010/12/06/arduino-lezione-03-controlliamo-un-led-con-un-pulsante/?replytocom=1066#respond)