**Arduino lezione 04 realizzare un programma che identifica le variazioni di stato**

Pubblicato il [marzo 10, 2011](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/) da [admin](http://www.maffucci.it/author/admin/" \o "Visualizza tutti gli articoli di admin)

Prenderò spunto dal commento che mi è stato lasciato da un mio studente per introdurre l’uso delle istruzioni:

if

if...else

Switch Case

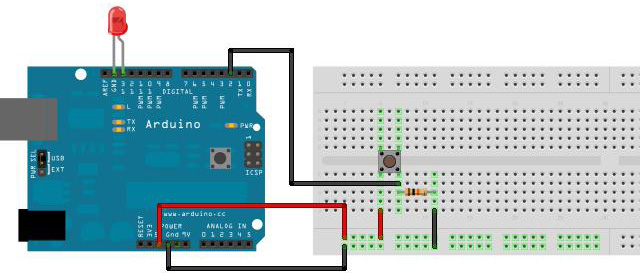
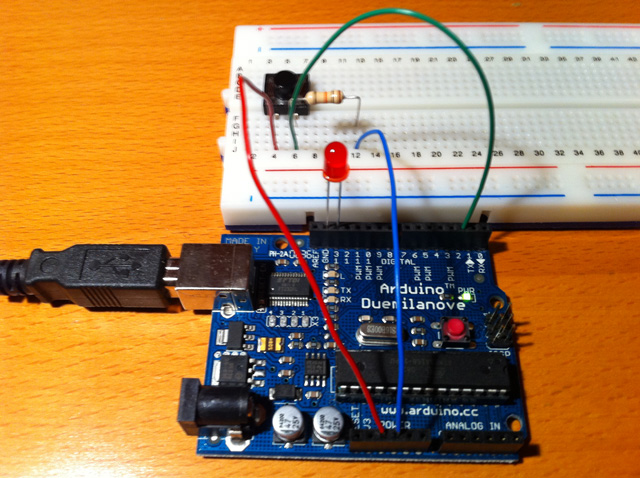
Lo scopo di questa lezione è quello di realizzare un programma che conta quante volte il pulsante viene premuto, per un ciclo di 4 pressioni, alla quinta pressione il led lampeggia una volta e poi si riavvia il ciclo.

Per questa lezione useremo:

1. breadboard  
2. 4 diodi led  
3. 1 pulsante  
4. 1 resistenza da 10KOhm  
5. Arduino duemilanove

La prima versione del programma è quella che è stata realizzata dal mio studente, realizzazione di uno sketch per effettuare la seguente funzione:

Prima pressione: led lampeggia ogni mezzo secondo  
Seconda pressione: led lampeggia ogni secondo  
Terza pressione: led lampeggia ogni secondo e mezzo  
Quarta pressione: led si spegne

Realizziamo un circuito come indicato nelle immagini che seguono:  
  


La proposta interessante ed elegante dell’allievo, sfrutta l’istruzione “**Switch Case**” che permette di definire quale porzione di codice eseguire in funzione del valore di una variabile. Ciò consente di evitare lunghe catene di if-else.  
Il “**break**” alla fine di ogni “**case**” viene usato per interrompere il controllo degli altri “**case**“.

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. // Esempio 05:
2. // Primo tocco: led lampeggia ogni mezzo secondo
3. // Secondo tocco: led lampeggia ogni secondo
4. // Terzo tocco: led lampeggia ogni secondo e mezzo
5. // Quarto tocco: led si spegne
7. **const** **int**  BUTTON = 2;    // pin di input a cui è collegato il pulsante
8. **const** **int** LED = 13;       // LED collegato al pin digitale 13
10. // Variabili globali (tutti interi)
11. **int** statoButton     = 0;      // stato del pulsante (inizialmente non premuto)
12. **int** lastStatoButton = 0;      // ultimo stato del pulsante (per ora non premuto)
13. **int** countButton     = 0;      // Conteggio del bottone
15. // Avvio dell'applicazione
16. **void** setup()
17. {
18. pinMode(LED, OUTPUT);       // imposta il pin digitale come output
19. pinMode(BUTTON, INPUT);     // imposta il pin digitale come input
20. }
22. // Avvio del loop
23. **void** loop()
24. {
26. // Verifico se l'utente ha premuto il bottone
27. **if**(digitalRead(BUTTON))
28. {
29. // Aspetto 15ms per far alzare il dito all'utente
30. delay(15);
31. // Cambio l'ultimo stato del bottone
32. **if**(lastStatoButton==0) lastStatoButton=1;
33. **else** lastStatoButton=0;
34. // Aumento il count del bottone
35. **if**(countButton<=3) countButton=countButton+1;
36. **else** countButton=0;
37. }
39. // In base allo stato del bottone scelgo l'azione del led
40. **switch** (countButton)
41. {
42. // Led lampeggia ogni mezzo secondo
43. **case** 1:
44. digitalWrite(LED, HIGH);  // accende il LED
45. delay(500);              // aspetta un secondo
46. digitalWrite(LED, LOW);   // spegne il LED
47. delay(500);              // aspetta un secondo
48. **break**;
50. // Led lampeggia ogni secondo
51. **case** 2:
52. digitalWrite(LED, HIGH);  // accende il LED
53. delay(1000);              // aspetta un secondo
54. digitalWrite(LED, LOW);   // spegne il LED
55. delay(1000);              // aspettaun secondo
56. **break**;
58. // led lampeggia ogni secondo e mezzo
59. **case** 3:
60. digitalWrite(LED, HIGH);  // accende il LED
61. delay(1500);              // aspetta un secondo
62. digitalWrite(LED, LOW);   // spegne il LED
63. delay(1500);              // aspetta un secondo
64. **break**;
66. // Led si spegne
67. **case** 0:
68. delay(15);
69. digitalWrite(LED, LOW);
70. delay(5000);              // aspetta un secondo
71. **break**;
72. }
73. }

Il programma come si intuisce dal filmato, presenta alcuni problemi.

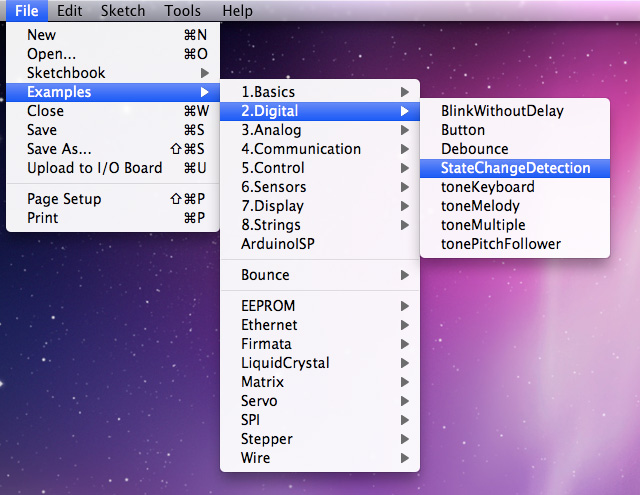
**Rimbalzo**  
Il problema dei segnali spuri dovuti al rimbalzo del pulsante non consente di controllare esattamente la selezione.

**Attesa**  
Durante l’attesa, mentre si vede lampeggiare il led, premendo il pulsante non si ha cambiamento di stato fino a quando non termina il tempo in cui lampeggia il led.

**Discriminare**  
Difficoltà di discriminare in quale situazione ci si trova, tempi troppo brevi per percepire la differenza delle 4 condizioni.

Per risolvere i problemi sopra elencati vi propongo 6 varianti all’esempio 1, ciò mi permetterà di introdurre nuove istruzioni del linguaggio di programmazione.

Prima di fornirvi le mie soluzioni ho necessità di introdurre l’uso dell’istruzioni Serial ed utilizzare il programma “StateChangeDetection” che potete trovare in **File > Examples > 2.Digitale > StateChangeDetection** o direttamente on-line sul [sito Arduino](http://arduino.cc/en/Tutorial/ButtonStateChange) che trovate di seguito

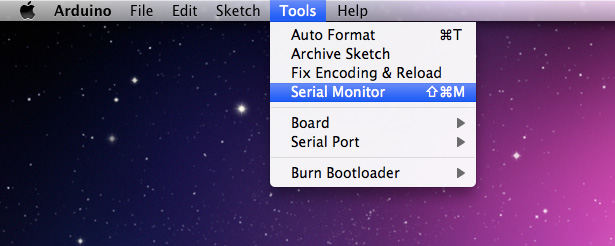


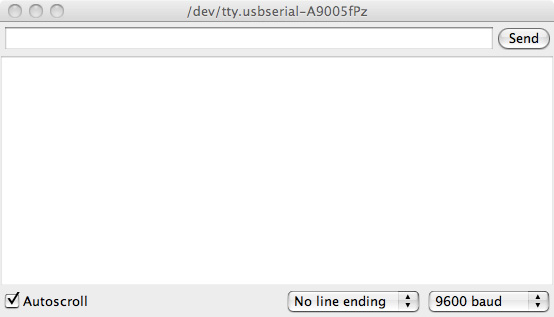
[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. /\*  State change detection (edge detection)
3. Often, you don't need to know the state of a digital input all the time,
4. but you just need to know when the input changes from one state to another.
5. For example, you want to know when a button goes from OFF to ON.  This is called
6. state change detection, or edge detection.
8. This example shows how to detect when a button or button changes from off to on
9. and on to off.
11. The circuit:
12. \* pushbutton attached to pin 2 from +5V
13. \* 10K resistor attached to pin 2 from ground
14. \* LED attached from pin 13 to ground (or use the built-in LED on
15. most Arduino boards)
17. created  27 Sep 2005
18. modified 30 Dec 2009
19. by Tom Igoe
21. This example code is in the public domain.
23. http://arduino.cc/en/Tutorial/ButtonStateChange
25. \*/
27. // this constant won't change:
28. **const** **int**  buttonPin = 2;    // the pin that the pushbutton is attached to
29. **const** **int** ledPin = 13;       // the pin that the LED is attached to
31. // Variables will change:
32. **int** buttonPushCounter = 0;   // counter for the number of button presses
33. **int** buttonState = 0;         // current state of the button
34. **int** lastButtonState = 0;     // previous state of the button
36. **void** setup() {
37. // initialize the button pin as a input:
38. pinMode(buttonPin, INPUT);
39. // initialize the LED as an output:
40. pinMode(ledPin, OUTPUT);
41. // initialize serial communication:
42. Serial.begin(9600);
43. }
45. **void** loop() {
46. // read the pushbutton input pin:
47. buttonState = digitalRead(buttonPin);
49. // compare the buttonState to its previous state
50. **if** (buttonState != lastButtonState) {
51. // if the state has changed, increment the counter
52. **if** (buttonState == HIGH) {
53. // if the current state is HIGH then the button
54. // wend from off to on:
55. buttonPushCounter++;
56. Serial.println("on");
57. Serial.print("number of button pushes:  ");
58. Serial.println(buttonPushCounter, DEC);
59. }
60. **else** {
61. // if the current state is LOW then the button
62. // wend from on to off:
63. Serial.println("off");
64. }
66. // save the current state as the last state,
67. //for next time through the loop
68. lastButtonState = buttonState;
69. }
71. // turns on the LED every four button pushes by
72. // checking the modulo of the button push counter.
73. // the modulo function gives you the remainder of
74. // the division of two numbers:
75. **if** (buttonPushCounter % 4 == 0) {
76. digitalWrite(ledPin, HIGH);
77. } **else** {
78. digitalWrite(ledPin, LOW);
79. }
81. }

Il programma utilizza l’istruzione “[Serial](http://arduino.cc/en/Reference/Serial)” usata per comunicare tra Arduino e il computer oppure altri dispositivi. La comunicazione tra Arduino e dispositivi esterni tramite la porta USB oppure tramite i pin 0 (RX) e 1 (TX). Ovviamente se si utilizzano i pin 1 e 0 per la comunicazione esterna, questi non potranno essere usati come input e output digitali.

All’interno dell’ambiente di sviluppo è implementata una shell per la comunicazione con la scheda, la trovate in **Tools > Serial Monitor** apparirà una finestra come quella rappresentata di seguito, in cui dovete specificare lo stesso baud rate che viene specificato nel [begin()](http://arduino.cc/en/Serial/Begin).





Tornando al programma StateChangeDetection si noti

Serial.begin(9600);

con cui apriamo la porta seriale USB e la inizializziamo a 9600 bps.

Nel programma troviamo l’istruzione:

Serial.println("on");

[println()](http://arduino.cc/en/Serial/Println) stampa sulla console di uscita il testo specificato tra parentesi e tra virgolette: “on” seguito da un return (ASCII 13, o ‘\r’) e una nuova linea (ASCII 10, or ‘\n’).

Serial.print("number of button pushes: ");

[print()](http://arduino.cc/en/Serial/Print) nel nostro caso stampa sulla console di uscita il testo specificato tra parentesi e tra virgolette: “number of button pushes: “.

Dal reference on-line di Arduino  
Con print() i numeri sono stampati utilizzando un carattere ASCII per ogni cifra, i float allo stesso modo ma con solo due numeri decimali, i byte vengono stampati come singoli caratteri, caratteri e stringhe vengono stampate senza nessuna conversione.

Esempio:

Serial.print(78) stampa “78″  
Serial.print(1.23456) stampa “1.23″  
Serial.print(byte(78)) stampa “N” (whose ASCII value is 78)  
Serial.print(‘N’) stampa “N”  
Serial.print(“Hello world.”) stampa “Hello world.”

print() può avere un secondo parametro che indica il formato usato nel primo parametro. Il formato può essere: BYTE, BIN (binario), OCT (ottale), DEC (decimale), HEX (esadecimale). Per i numeri in floating point il secondo parametro indica il numero di decimali presi in consideraione.

Per esempio:

Serial.print(78, BYTE) stampa “N”  
Serial.print(78, BIN) stampa “1001110″  
Serial.print(78, OCT) stampa “116″  
Serial.print(78, DEC) stampa “78″  
Serial.print(78, HEX) stampa “4E”  
Serial.println(1.23456, 0) stampa “1″  
Serial.println(1.23456, 2) stampa “1.23″  
Serial.println(1.23456, 4) stampa “1.2346″

Torniamo all’analisi del programma CountStateDetection

buttonState = digitalRead(buttonPin);

buttonState = digitalRead(buttonPin);

immagazziniamo in buttonState lo stato del pulsante

if (buttonState != lastButtonState) {

...

controlliamo se il pulsante è stato premuto, cioè se lo stato attuale del pulsante è diverso da quello precedente, se la condizione è vera viene eseguita la porzione di codice che segue:

…  
// if the state has changed, increment the counter  
if (buttonState == HIGH) {  
// if the current state is HIGH then the button  
// wend from off to on:  
buttonPushCounter++;  
Serial.println(“on”);  
Serial.print(“number of button pushes: “);  
Serial.println(buttonPushCounter, DEC);  
}  
else {  
// if the current state is LOW then the button  
// wend from on to off:  
Serial.println(“off”);  
}  
…

se buttonState è HIGH vuol dire che si è passati da pulsante non premuto a pulsante premuto,

...

if (buttonState == HIGH) {

...

incrementiamo di una unità la variabile che conta il numero di volte in cui il pulsante viene premuto

buttonPushCounter++;

si noti che la notazione sopra è analoga alla seguente:

buttonPushCounter = buttonPushCounter+1;

Successivamente con la porzione di codice:

...

Serial.println("on");

Serial.print("number of button pushes: ");

Serial.println(buttonPushCounter, DEC);

...

stampiamo sul serial monitor il testo “on”, si va a capo, si stampa: “number of button pushes: ” seguito dalla stampa del numero di volte in cui il pulsante è stato premuto, dopo di che si va a capo.

Altrimenti se la condizione dell’if è falsa, cioè il pulsante non è premuto, viene eseguita la parte di codice dell’else e viene stampato “off”

...

else {

// if the current state is LOW then the button

// wend from on to off:

Serial.println("off");

}

...

Il passo successivo è quello di immagazzinare lo stato del pulsante in lastButtonState.

Nel caso in cui la condizione del primo if del loop() risulta falso, cioè il pulsante non è stato premuto, sarà eseguita la porzione di codice:

...

// turns on the LED every four button pushes by

// checking the modulo of the button push counter.

// the modulo function gives you the remainder of

// the division of two numbers:

if (buttonPushCounter % 4 == 0) {

digitalWrite(ledPin, HIGH);

} else {

digitalWrite(ledPin, LOW);

}

...

Si controlla se la divisione intera per 4 del contatore di pressione pulsante da resto zero, se vero ciò vuol dire che abbiamo premuto per quattro volte il pulsante, se falso, si esegue il codice nell’else, ciò vuol dire che non abbiamo premuto per 4 volte il pulsante ed il led viene mantenuto spento.  
Ogni volta che si raggiungono 4 pressioni, si accende il led.

Ora incominciamo a dare soluzione al problema che ci siamo posti:  
realizzare un programma che conta quante volte il pulsante viene premuto, per un ciclo di 4 pressioni, alla quinta pressione il led lampeggia una volta e poi si riavvia il ciclo.

La prima soluzione sfrutta una serie di if annidati:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. // Esempio 06: conta quante volte il pulsante viene premuto, per un ciclo di 4 pressioni,
2. // alla quinta pressione il led lampeggia una volta e poi si riavvia il ciclo
4. #define BUTTON 2                                  // pin di input a cui è collegato il pulsante
5. #define LED 13                                    // LED collegato al pin digitale 13
7. // Variabili
8. **int** ContatorePulsantePremuto = 0;                   // conta il numero di volte che il pulsante è premuto buttonPushCounter
9. **int** StatoPulsante = 0;                              // stato corrente del pulsante
10. **int** StatoPulsantePrecedente = 0;                    // stato precedente del pulsante
12. **void** setup() {
13. pinMode(BUTTON, INPUT);                           // imposta il pin digitale come output
14. pinMode(LED, OUTPUT);                             // imposta il pin digitale come input
15. Serial.begin(9600);                               // apre la porta seriale e la inizzializza a 9600 bps
16. }
18. **void** loop() {
19. StatoPulsante = digitalRead(BUTTON);              // legge il valore dell'input e lo conserva
21. **if** (StatoPulsante != StatoPulsantePrecedente) {   // compara lo stato del pulsante attuale con il precedente
22. **if** (StatoPulsante == HIGH) {                    // se lo stato è cambiato incrementa il contatore
23. // se lo stato corrente è alto, il pulsante è passato da off a on
24. ContatorePulsantePremuto++;
25. **if** (ContatorePulsantePremuto == 1) {                   // controlla se il pulsante è stato premuto una volta
26. Serial.println("on");                                // stampa sulla console "on"
27. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
28. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
29. Serial.println("off");                               // stampa sulla console "off"
30. }
31. **if** (ContatorePulsantePremuto == 2) {                   // controlla se il pulsante è stato premuto due volte
32. Serial.println("on");
33. Serial.print("numero di volte tasto premuto:  ");
34. Serial.println(ContatorePulsantePremuto, DEC);
35. Serial.println("off");
36. }
37. **if** (ContatorePulsantePremuto == 3) {                   // controlla se il pulsante è stato premuto tre volte
38. Serial.println("on");
39. Serial.print("numero di volte tasto premuto:  ");
40. Serial.println(ContatorePulsantePremuto, DEC);
41. Serial.println("off");
42. }
43. **if** (ContatorePulsantePremuto == 4) {                   // controlla se il pulsante è stato premuto quattro volte
44. Serial.println("on");
45. Serial.print("numero di volte tasto premuto:  ");
46. Serial.println(ContatorePulsantePremuto, DEC);
47. Serial.println("off");
48. }
49. }
50. }
52. // salva lo stato corrente nella variabile che indica lo stato precedente per il loop successivo
53. StatoPulsantePrecedente = StatoPulsante;
55. // controlla se il pulsante è stato premuto quattro volte se vero indica che è finito il ciclo
56. // il led lampeggia 2 volte per 50 millisecondi
57. // vengono inizializzate nuovamente le variabili
58. // si riavvia il ciclo
60. **if** (ContatorePulsantePremuto > 4) {
61. Serial.println("fine ciclo");
62. digitalWrite(LED, HIGH);                                 // accende il LED
63. delay(50);                                               // aspetta 50 millisecondi
64. digitalWrite(LED, LOW);                                  // spegne il LED
65. delay(50);                                               // aspetta 50 millisecondi
66. // inizializzazione delle variabili
67. ContatorePulsantePremuto = 0;
68. StatoPulsante = 0;
69. StatoPulsantePrecedente = 0;
70. Serial.println("mi riavvio");                            // stampa sulla console "mi riavvio"
71. }
72. }

come potete notare il programma è molto simile a CountStateDetection ma in questo caso verifico con una serie di if se il pulsante è stato premuto 1,2,3,4 volte,

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. ...
2. **if** (ContatorePulsantePremuto == 1) {                   // controlla se il pulsante è stato premuto una volta
3. Serial.println("on");                                // stampa sulla console "on"
4. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
5. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
6. Serial.println("off");                               // stampa sulla console "off"
7. }
8. **if** (ContatorePulsantePremuto == 2) {                   // controlla se il pulsante è stato premuto due volte
9. Serial.println("on");
10. Serial.print("numero di volte tasto premuto:  ");
11. Serial.println(ContatorePulsantePremuto, DEC);
12. Serial.println("off");
13. }
14. **if** (ContatorePulsantePremuto == 3) {                   // controlla se il pulsante è stato premuto tre volte
15. Serial.println("on");
16. Serial.print("numero di volte tasto premuto:  ");
17. Serial.println(ContatorePulsantePremuto, DEC);
18. Serial.println("off");
19. }
20. **if** (ContatorePulsantePremuto == 4) {                   // controlla se il pulsante è stato premuto quattro volte
21. Serial.println("on");
22. Serial.print("numero di volte tasto premuto:  ");
23. Serial.println(ContatorePulsantePremuto, DEC);
24. Serial.println("off");
25. }
26. ...

alla quinta pressione, scriviamo a monitor: “fine ciclo”, facciamo lampeggiare il led per 50 millisecondi, stampiamo a monitor “mi riavvio” e si incomincia nuovamente il ciclo.

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. ...
2. // controlla se il pulsante è stato premuto quattro volte se vero indica che è finito il ciclo
3. // il led lampeggia 2 volte per 50 millisecondi
4. // vengono inizializzate nuovamente le variabili
5. // si riavvia il ciclo
7. **if** (ContatorePulsantePremuto > 4) {
8. Serial.println("fine ciclo");
9. digitalWrite(LED, HIGH);                                 // accende il LED
10. delay(50);                                               // aspetta 50 millisecondi
11. digitalWrite(LED, LOW);                                  // spegne il LED
12. delay(50);                                               // aspetta 50 millisecondi
13. // inizializzazione delle variabili
14. ContatorePulsantePremuto = 0;
15. StatoPulsante = 0;
16. StatoPulsantePrecedente = 0;
17. Serial.println("mi riavvio");                            // stampa sulla console "mi riavvio"
18. }
19. ...

Poichè la visualizzazione dell’accensione del led alla quinta pressione del pulsante è troppo breve, facciamo in modo che alla quinta prssione il led lampeggi per 10 volte ad intervalli di 50 millisecondi, per far ciò utilizzerò un ciclo **for** ([vai al reference on-line per la spiegazione](http://arduino.cc/en/Reference/For)).

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. // Esempio 07: conta quante volte il pulsante viene premuto, per un ciclo di 4 pressioni,
2. // alla quinta pressione il led lampeggia per 10 volte e poi si riavvia il ciclo
3. // per il ciclo lampeggio è stato utilizzato un ciclo for
5. #define BUTTON 2                                  // pin di input a cui è collegato il pulsante
6. #define LED 13                                    // LED collegato al pin digitale 13
8. // Variabili
9. **int** ContatorePulsantePremuto = 0;                   // conta il numero di volte che il pulsante è premuto buttonPushCounter
10. **int** StatoPulsante = 0;                              // stato corrente del pulsante
11. **int** StatoPulsantePrecedente = 0;                    // stato precedente del pulsante
13. **void** setup() {
14. pinMode(BUTTON, INPUT);                           // imposta il pin digitale come input
15. pinMode(LED, OUTPUT);                             // imposta il pin digitale come output
16. Serial.begin(9600);                               // apre la porta seriale e la inizzializza a 9600 bps
17. }
19. **void** loop() {
20. StatoPulsante = digitalRead(BUTTON);              // legge il valore dell'input e lo conserva
22. **if** (StatoPulsante != StatoPulsantePrecedente) {   // compara lo stato del pulsante attuale con il precedente
23. **if** (StatoPulsante == HIGH) {                    // se lo stato è cambiato incrementa il contatore
24. // se lo stato corrente è alto, il pulsante è passato da off a on
25. ContatorePulsantePremuto++;
26. **if** (ContatorePulsantePremuto == 1) {                   // controlla se il pulsante è stato premuto una volta
27. Serial.println("on");                                // stampa sulla console "on"
28. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
29. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
30. Serial.println("off");                               // stampa sulla console "off"
31. }
32. **if** (ContatorePulsantePremuto == 2) {                   // controlla se il pulsante è stato premuto due volte
33. Serial.println("on");
34. Serial.print("numero di volte tasto premuto:  ");
35. Serial.println(ContatorePulsantePremuto, DEC);
36. Serial.println("off");
37. }
38. **if** (ContatorePulsantePremuto == 3) {                   // controlla se il pulsante è stato premuto tre volte
39. Serial.println("on");
40. Serial.print("numero di volte tasto premuto:  ");
41. Serial.println(ContatorePulsantePremuto, DEC);
42. Serial.println("off");
43. }
44. **if** (ContatorePulsantePremuto == 4) {                   // controlla se il pulsante è stato premuto quattro volte
45. Serial.println("on");
46. Serial.print("numero di volte tasto premuto:  ");
47. Serial.println(ContatorePulsantePremuto, DEC);
48. Serial.println("off");
49. }
50. }
51. }
53. // salva lo stato corrente nella variabile che indica lo stato precedente per il loop successivo
54. StatoPulsantePrecedente = StatoPulsante;
56. // controlla se il pulsante è stato premuto quattro volte se vero indica che è finito il ciclo
57. // il led lampeggia 2 volte per 50 millisecondi
58. // vengono inizializzate nuovamente le variabili
59. // si riavvia il ciclo
61. **if** (ContatorePulsantePremuto > 4) {
62. Serial.println("fine ciclo");
63. **for** (**int** x=0; x<10; x++) {                                 // lampeggia per 10 volte
64. digitalWrite(LED, HIGH);                                 // accende il LED
65. delay(50);                                               // aspetta 50 millisecondi
66. digitalWrite(LED, LOW);                                  // spegne il LED
67. delay(50);                                               // aspetta 50 millisecondi
68. }
69. // inizializzazione delle variabili
70. ContatorePulsantePremuto = 0;
71. StatoPulsante = 0;
72. StatoPulsantePrecedente = 0;
73. Serial.println("mi riavvio");                            // stampa sulla console "mi riavvio"
74. }
75. }

La parte di codice che consente di far lampeggiare 10 volte il led è:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. ...
2. **for** (**int** x=0; x<10; x++) {                                 // lampeggia per 10 volte
3. digitalWrite(LED, HIGH);                                 // accende il LED
4. delay(50);                                               // aspetta 50 millisecondi
5. digitalWrite(LED, LOW);                                  // spegne il LED
6. delay(50);                                               // aspetta 50 millisecondi
7. }
8. ...

Si inizializza la viariabile intera x a 0 e ad ogni ciclo la si incrementa di 1 (x++)

Nel programma precedente, il controllo delle volte in cui il pulsante viene premuto è effettuato dalla serie di if:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

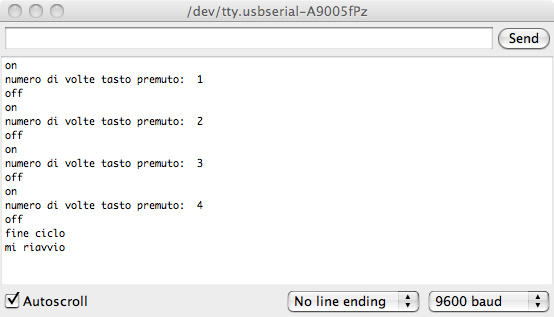
1. ...
2. **if** (ContatorePulsantePremuto == 1) {                   // controlla se il pulsante è stato premuto una volta
3. Serial.println("on");                                // stampa sulla console "on"
4. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
5. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
6. Serial.println("off");                               // stampa sulla console "off"
7. }
8. **if** (ContatorePulsantePremuto == 2) {                   // controlla se il pulsante è stato premuto due volte
9. Serial.println("on");
10. Serial.print("numero di volte tasto premuto:  ");
11. Serial.println(ContatorePulsantePremuto, DEC);
12. Serial.println("off");
13. }
14. **if** (ContatorePulsantePremuto == 3) {                   // controlla se il pulsante è stato premuto tre volte
15. Serial.println("on");
16. Serial.print("numero di volte tasto premuto:  ");
17. Serial.println(ContatorePulsantePremuto, DEC);
18. Serial.println("off");
19. }
20. **if** (ContatorePulsantePremuto == 4) {                   // controlla se il pulsante è stato premuto quattro volte
21. Serial.println("on");
22. Serial.print("numero di volte tasto premuto:  ");
23. Serial.println(ContatorePulsantePremuto, DEC);
24. Serial.println("off");
25. }
26. ...

dal punto di vista della programmazione ritengo più elegante e chiaro utilizzare l’istruzione**case** spiegato ad inizio lezione, questo è il codice:

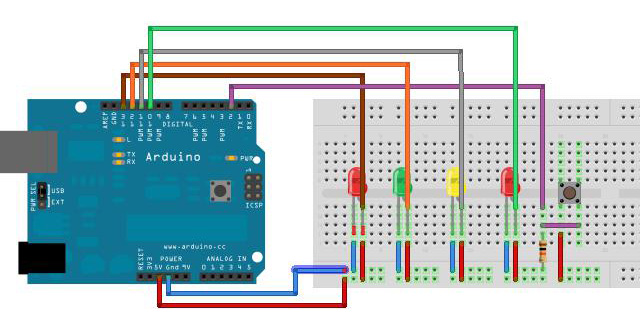
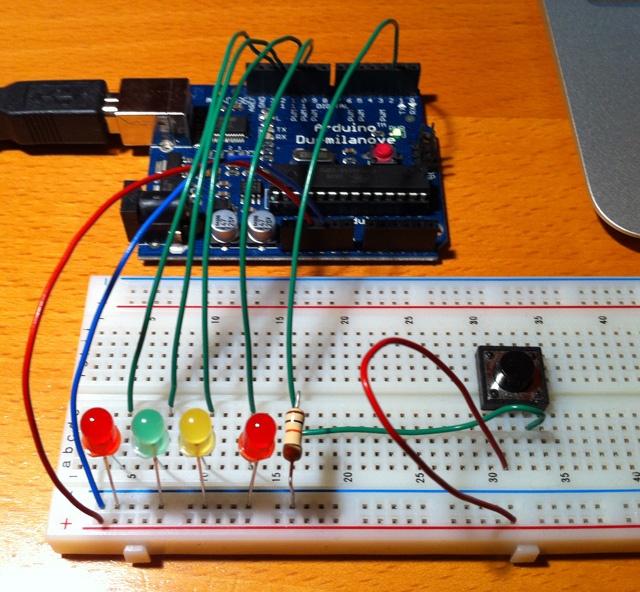
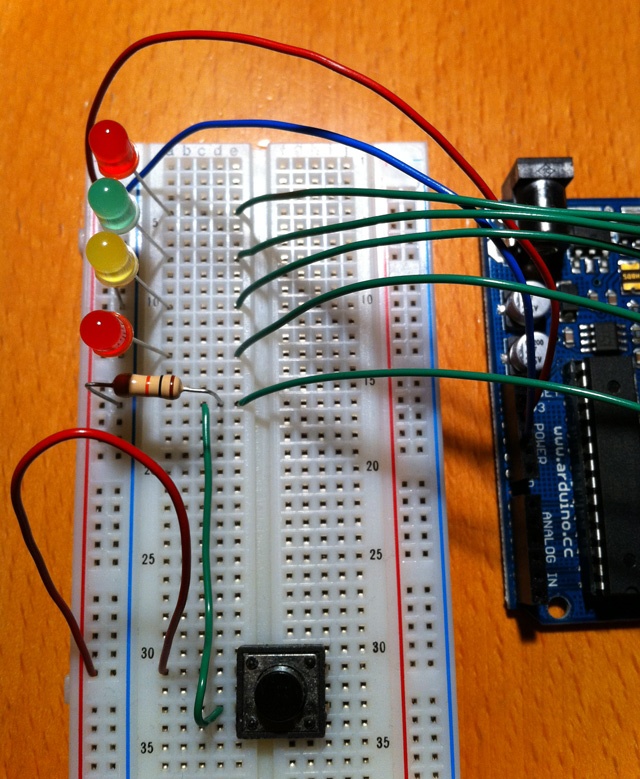
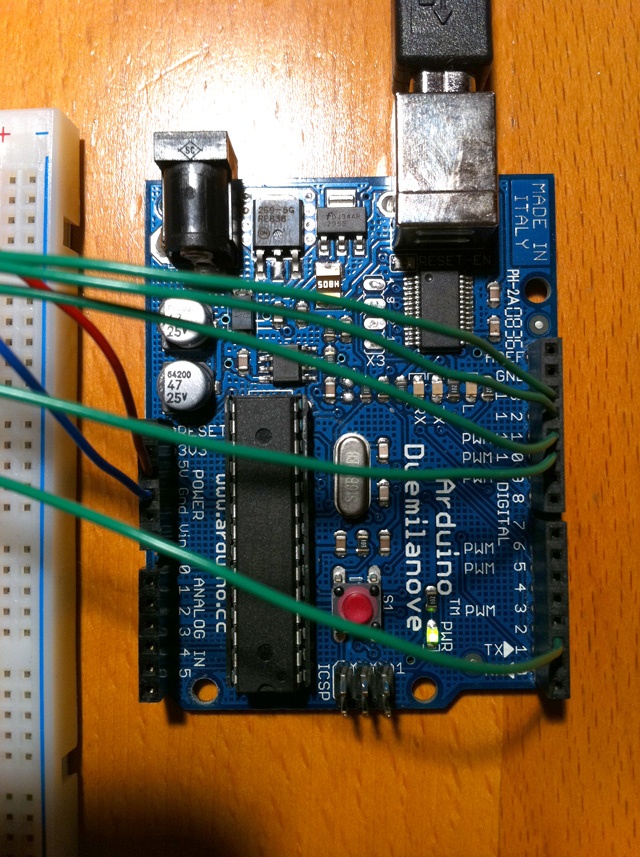
[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. // Esempio 08: conta quante volte il pulsante viene premuto, per un ciclo di 4 pressioni,
2. // alla quinta pressione il led lampeggia per 10 volte e poi si riavvia il ciclo.
3. // Il controllo del numero di volte in cui il pulsante viene premuto viene fatto con l'istruzione "case"
5. #define BUTTON 2                                  // pin di input a cui è collegato il pulsante
6. #define LED 13                                    // LED collegato al pin digitale 13
8. // Variabili
9. **int** ContatorePulsantePremuto = 0;                   // conta il numero di volte che il pulsante è premuto buttonPushCounter
10. **int** StatoPulsante = 0;                              // stato corrente del pulsante
11. **int** StatoPulsantePrecedente = 0;                    // stato precedente del pulsante
13. **void** setup() {
14. pinMode(BUTTON, INPUT);                           // imposta il pin digitale come input
15. pinMode(LED, OUTPUT);                             // imposta il pin digitale come output
16. Serial.begin(9600);                               // apre la porta seriale e la inizzializza a 9600 bps
17. }
19. **void** loop() {
20. StatoPulsante = digitalRead(BUTTON);              // legge il valore dell'input e lo conserva
22. **if** (StatoPulsante != StatoPulsantePrecedente) {   // compara lo stato del pulsante attuale con il precedente
23. **if** (StatoPulsante == HIGH) {                    // se lo stato è cambiato incrementa il contatore
24. // se lo stato corrente è alto, il pulsante è passato da off a on
25. ContatorePulsantePremuto++;
27. **switch** (ContatorePulsantePremuto) {
28. **case** 1:  // controlla se il pulsante è stato premuto una volta
29. Serial.println("on");                                // stampa sulla console "on"
30. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
31. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
32. Serial.println("off");                               // stampa sulla console "off"
33. **break**;
34. **case** 2:  // controlla se il pulsante è stato premuto due volte
35. Serial.println("on");                                // stampa sulla console "on"
36. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
37. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
38. Serial.println("off");                               // stampa sulla console "off"
39. **break**;
40. **case** 3:  // controlla se il pulsante è stato premuto tre volte
41. Serial.println("on");                                // stampa sulla console "on"
42. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
43. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
44. Serial.println("off");                               // stampa sulla console "off"
45. **break**;
46. **case** 4:  // controlla se il pulsante è stato premuto quattro volte
47. Serial.println("on");                                // stampa sulla console "on"
48. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
49. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
50. Serial.println("off");                               // stampa sulla console "off"
51. **break**;
52. }
53. }
54. }
56. // salva lo stato corrente nella variabile che indica lo stato precedente per il loop successivo
57. StatoPulsantePrecedente = StatoPulsante;
59. // controlla se il pulsante è stato premuto quattro volte se vero indica che è finito il ciclo
60. // il led lampeggia 2 volte per 50 millisecondi
61. // vengono inizializzate nuovamente le variabili
62. // si riavvia il ciclo
64. **if** (ContatorePulsantePremuto > 4) {
65. Serial.println("fine ciclo");
66. **for** (**int** x=0; x<10; x++) {                                 // lampeggia per 10 volte
67. digitalWrite(LED, HIGH);                                 // accende il LED
68. delay(50);                                               // aspetta 50 millisecondi
69. digitalWrite(LED, LOW);                                  // spegne il LED
70. delay(50);                                               // aspetta 50 millisecondi
71. }
72. // inizializzazione delle variabili
73. ContatorePulsantePremuto = 0;
74. StatoPulsante = 0;
75. StatoPulsantePrecedente = 0;
76. Serial.println("mi riavvio");                            // stampa sulla console "mi riavvio"
77. }
78. }

L’esercizio che segue aggiunge un grado di difficoltà.  
Negli esempi precedenti per controllare il numero di volte che il pulsante è stato premuto si è inviato un messaggio alla shell di output (Serial Monitor) se ricordate dai filmati l’output era questo:



Vogliamo ora, oltre che mandare messaggi alla shell di output, usare 4 led per identificare la pressione del pulsante.  
Ad ogni pressione del pulsante si farà accende un led per volta e alla quinta pressione il primo led lampeggia per 10 volte.  
Anche in questo caso il controllo del numero di volte in cui il pulsante viene premuto viene fatto con l’istruzione “case”.

Questo il circuito:  
  
  
  


Il programma:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. // Esempio 9: conta quante volte il pulsante viene premuto, per un ciclo di 4 pressioni,
2. // ad ogni pressione del pulsante si accende un led per volta
3. // alla quinta pressione il primo led lampeggia per 10 volte.
4. // Il controllo del numero di volte in cui il pulsante viene premuto viene fatto con l'istruzione "case"
6. #define BUTTON 2                                   // pin di input a cui è collegato il pulsante
7. #define LED1 13                                    // LED collegato al pin digitale 13
8. #define LED2 12                                    // LED collegato al pin digitale 12
9. #define LED3 11                                    // LED collegato al pin digitale 11
10. #define LED4 10                                    // LED collegato al pin digitale 10
12. // Variabili
13. **int** ContatorePulsantePremuto = 0;                   // conta il numero di volte che il pulsante è premuto buttonPushCounter
14. **int** StatoPulsante = 0;                              // stato corrente del pulsante
15. **int** StatoPulsantePrecedente = 0;                    // stato precedente del pulsante
17. **void** setup() {
18. pinMode(BUTTON, INPUT);                           // imposta il pin digitale come output
19. pinMode(LED1, OUTPUT);                            // imposta il pin digitale come input
20. pinMode(LED2, OUTPUT);                            // imposta il pin digitale come input
21. pinMode(LED3, OUTPUT);                            // imposta il pin digitale come input
22. pinMode(LED4, OUTPUT);                            // imposta il pin digitale come input
23. Serial.begin(9600);                               // apre la porta seriale e la inizzializza a 9600 bps
24. }
26. **void** loop() {
27. StatoPulsante = digitalRead(BUTTON);              // legge il valore dell'input e lo conserva
29. **if** (StatoPulsante != StatoPulsantePrecedente) {   // compara lo stato del pulsante attuale con il precedente
30. **if** (StatoPulsante == HIGH) {                    // se lo stato è cambiato incrementa il contatore
31. // se lo stato corrente è alto, il pulsante è passato da off a on
32. ContatorePulsantePremuto++;
34. **switch** (ContatorePulsantePremuto) {
35. **case** 1:  // controlla se il pulsante è stato premuto una volta
36. Serial.println("on");                                // stampa sulla console "on"
37. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
38. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
39. digitalWrite(LED1, HIGH);                                 // accende il LED
40. Serial.println("off");                               // stampa sulla console "off"
41. **break**;
42. **case** 2:  // controlla se il pulsante è stato premuto due volte
43. Serial.println("on");                                // stampa sulla console "on"
44. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
45. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
46. digitalWrite(LED1, LOW);                                 // accende il LED
47. digitalWrite(LED2, HIGH);                                 // accende il LED
48. Serial.println("off");                               // stampa sulla console "off"
49. **break**;
50. **case** 3:  // controlla se il pulsante è stato premuto tre volte
51. Serial.println("on");                                // stampa sulla console "on"
52. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
53. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
54. digitalWrite(LED2, LOW);                                 // accende il LED
55. digitalWrite(LED3, HIGH);                                 // accende il LED
56. Serial.println("off");                               // stampa sulla console "off"
57. **break**;
58. **case** 4:  // controlla se il pulsante è stato premuto quattro volte
59. Serial.println("on");                                // stampa sulla console "on"
60. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
61. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
62. digitalWrite(LED3, LOW);                                 // accende il LED
63. digitalWrite(LED4, HIGH);                                 // accende il LED
64. Serial.println("off");                               // stampa sulla console "off"
65. **break**;
66. }
67. }
68. }
70. // salva lo stato corrente nella variabile che indica lo stato precedente per il loop successivo
71. StatoPulsantePrecedente = StatoPulsante;
73. // controlla se il pulsante è stato premuto quattro volte se vero indica che è finito il ciclo
74. // il led lampeggia 2 volte per 50 millisecondi
75. // vengono inizializzate nuovamente le variabili
76. // si riavvia il ciclo
78. **if** (ContatorePulsantePremuto > 4) {
79. Serial.println("fine ciclo");
80. digitalWrite(LED4, LOW);                                 // accende il LED
81. delay(50);
82. **for** (**int** x=0; x<10; x++) {                                 // lampeggia per 10 volte
83. digitalWrite(LED1, HIGH);                                 // accende il LED
84. delay(50);                                               // aspetta 50 millisecondi
85. digitalWrite(LED1, LOW);                                  // spegne il LED
86. delay(50);                                               // aspetta 50 millisecondi
87. }
88. // inizializzazione delle variabili
89. ContatorePulsantePremuto = 0;
90. StatoPulsante = 0;
91. StatoPulsantePrecedente = 0;
92. Serial.println("mi riavvio");                            // stampa sulla console "mi riavvio"
93. }
94. }

Il programma differisce dalla versione precedente sostanzialmente nella parte iniziale in cui nominiamo i vari pin con nome del led corrispondente:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. ...
2. #define LED1 13                                    // LED collegato al pin digitale 13
3. #define LED2 12                                    // LED collegato al pin digitale 12
4. #define LED3 11                                    // LED collegato al pin digitale 11
5. #define LED4 10                                    // LED collegato al pin digitale 10
6. ...

e nell’area dei “**case**” in cui come potete notare ad ogni pressione del pulsante viene spento il led precedente e acceso il successivo:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. ...
2. **case** 1:  // controlla se il pulsante è stato premuto una volta
3. Serial.println("on");                                // stampa sulla console "on"
4. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
5. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
7. digitalWrite(LED1, HIGH);                                 // accende il LED
9. Serial.println("off");                               // stampa sulla console "off"
10. **break**;
11. **case** 2:  // controlla se il pulsante è stato premuto due volte
12. Serial.println("on");                                // stampa sulla console "on"
13. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
14. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
16. digitalWrite(LED1, LOW);                                 // accende il LED
17. digitalWrite(LED2, HIGH);                                 // accende il LED
19. Serial.println("off");                               // stampa sulla console "off"
20. **break**;
21. **case** 3:  // controlla se il pulsante è stato premuto tre volte
22. Serial.println("on");                                // stampa sulla console "on"
23. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
24. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
26. digitalWrite(LED2, LOW);                                 // accende il LED
27. digitalWrite(LED3, HIGH);                                 // accende il LED
29. Serial.println("off");                               // stampa sulla console "off"
30. **break**;
31. **case** 4:  // controlla se il pulsante è stato premuto quattro volte
32. Serial.println("on");                                // stampa sulla console "on"
33. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
34. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
36. digitalWrite(LED3, LOW);                                 // accende il LED
37. digitalWrite(LED4, HIGH);                                 // accende il LED
39. Serial.println("off");                               // stampa sulla console "off"
40. **break**;
41. ...

Lo scopo dell’ultimo esercizio è quello di creare un annidamento di “**for**“.  
Anche in questo caso ad ogni pressione del pulsante si accende un led per volta però ora alla quinta pressione i led lampeggia per 5 volte in sequenza dal 4° al 1° led e poi si riavvia il ciclo.

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. // Esempio 10: conta quante volte il pulsante viene premuto, per un ciclo di 4 pressioni,
2. // ad ogni pressione del pulsante si accende un led per volta
3. // alla quinta pressione i led lampeggia per 5 volte in sequenza dal 4' al 1' led e poi si riavvia il ciclo.
4. // Il controllo del numero di volte in cui il pulsante viene premuto viene fatto con l'istruzione "case"
6. #define BUTTON 2                                    // pin di input a cui è collegato il pulsante
7. #define LED1 13                                     // LED collegato al pin digitale 13
8. #define LED2 12                                     // LED collegato al pin digitale 12
9. #define LED3 11                                     // LED collegato al pin digitale 11
10. #define LED4 10                                     // LED collegato al pin digitale 10
12. // Variabili
13. **int** ContatorePulsantePremuto = 0;                   // conta il numero di volte che il pulsante è premuto buttonPushCounter
14. **int** StatoPulsante = 0;                              // stato corrente del pulsante
15. **int** StatoPulsantePrecedente = 0;                    // stato precedente del pulsante
17. **void** setup() {
18. pinMode(BUTTON, INPUT);                           // imposta il pin digitale come input
19. pinMode(LED1, OUTPUT);                            // imposta il pin digitale come output
20. pinMode(LED2, OUTPUT);                            // imposta il pin digitale come output
21. pinMode(LED3, OUTPUT);                            // imposta il pin digitale come output
22. pinMode(LED4, OUTPUT);                            // imposta il pin digitale come output
23. Serial.begin(9600);                               // apre la porta seriale e la inizzializza a 9600 bps
24. }
26. **void** loop() {
27. StatoPulsante = digitalRead(BUTTON);              // legge il valore dell'input e lo conserva
29. **if** (StatoPulsante != StatoPulsantePrecedente) {   // compara lo stato del pulsante attuale con il precedente
30. **if** (StatoPulsante == HIGH) {                    // se lo stato è cambiato incrementa il contatore
31. // se lo stato corrente è alto, il pulsante è passato da off a on
32. ContatorePulsantePremuto++;
34. **switch** (ContatorePulsantePremuto) {
35. **case** 1:  // controlla se il pulsante è stato premuto una volta
36. Serial.println("on");                                // stampa sulla console "on"
37. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
38. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
39. digitalWrite(LED1, HIGH);                            // accende il LED1
40. Serial.println("off");                               // stampa sulla console "off"
41. **break**;
42. **case** 2:  // controlla se il pulsante è stato premuto due volte
43. Serial.println("on");                                // stampa sulla console "on"
44. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
45. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
46. digitalWrite(LED1, LOW);                             // spegne il LED1
47. digitalWrite(LED2, HIGH);                            // accende il LED2
48. Serial.println("off");                               // stampa sulla console "off"
49. **break**;
50. **case** 3:  // controlla se il pulsante è stato premuto tre volte
51. Serial.println("on");                                // stampa sulla console "on"
52. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
53. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
54. digitalWrite(LED2, LOW);                             // spegne il LED2
55. digitalWrite(LED3, HIGH);                            // accende il LED3
56. Serial.println("off");                               // stampa sulla console "off"
57. **break**;
58. **case** 4:  // controlla se il pulsante è stato premuto quattro volte
59. Serial.println("on");                                // stampa sulla console "on"
60. Serial.print("numero di volte tasto premuto:  ");    // stampa sulla console "numero di volte tasto premuto:"
61. Serial.println(ContatorePulsantePremuto, DEC);       // stampa il numero di volte che il pulsante è stato premuto
62. digitalWrite(LED3, LOW);                             // accende il LED3
63. digitalWrite(LED4, HIGH);                            // accende il LED4
64. Serial.println("off");                               // stampa sulla console "off"
65. **break**;
66. }
67. }
68. }
70. // salva lo stato corrente nella variabile che indica lo stato precedente per il loop successivo
71. StatoPulsantePrecedente = StatoPulsante;
73. // controlla se il pulsante è stato premuto quattro volte se vero indica che è finito il ciclo
74. // il led lampeggia 2 volte per 50 millisecondi
75. // vengono inizializzate nuovamente le variabili
76. // si riavvia il ciclo
78. **if** (ContatorePulsantePremuto > 4) {
79. Serial.println("fine ciclo");                            // stampa sulla console "fine ciclo"
80. digitalWrite(LED4, LOW);                                 // spegne il LED4
81. **for** (**int** x=0; x<5; x++) {                                // ciclo di accensione e spegnimento led
82. **for** (**int** n=10; n<14; n++) {                            // ciclo sui diodi da accendere e spegnere
83. digitalWrite(n, HIGH);                               // accende il LED
84. delay(50);                                           // aspetta 50 millisecondi
85. digitalWrite(n, LOW);                                // spegne il LED
86. delay(50);                                           // aspetta 50 millisecondi
87. }
88. }
89. // inizializzazione delle variabili
90. ContatorePulsantePremuto = 0;
91. StatoPulsante = 0;
92. StatoPulsantePrecedente = 0;
93. Serial.println("mi riavvio");                            // stampa sulla console "mi riavvio"
94. }
95. }

I due cicli “**for**” sono nella parte terminale del programma:

[view plaincopy to clipboardprint?](http://www.maffucci.it/2011/03/10/arduino-lezione-4-realizzare-un-programma-che-identifica-le-variazioni-di-stato/)

1. ...
2. **for** (**int** x=0; x<5; x++) {                                // ciclo di accensione e spegnimento led
3. **for** (**int** n=10; n<14; n++) {                            // ciclo sui diodi da accendere e spegnere
4. digitalWrite(n, HIGH);                               // accende il LED
5. delay(50);                                           // aspetta 50 millisecondi
6. digitalWrite(n, LOW);                                // spegne il LED
7. delay(50);                                           // aspetta 50 millisecondi
8. }
9. }

Il ciclo “**for**” interno eseguirà 4 cicli (per n che va da 10 a 13). Come potete notare la variabile intera n viene inizializzata a 10 perché bisognerà accendere e poi spegnere prima il led connesso al pin 10, e così fino al diodo led connesso al pin 13 e questa sequenza sarà eseguita per 5 volte così come imposto dal ciclo “**for**” più esterno che eseguirà 5 (da 0 a 4) sequenze di accensione dei 4 led.

Bene!

Ora sapete usare le istruzioni: if, case e for e annidamento di for e sapete anche come inviare messaggi da Arduino al computer.