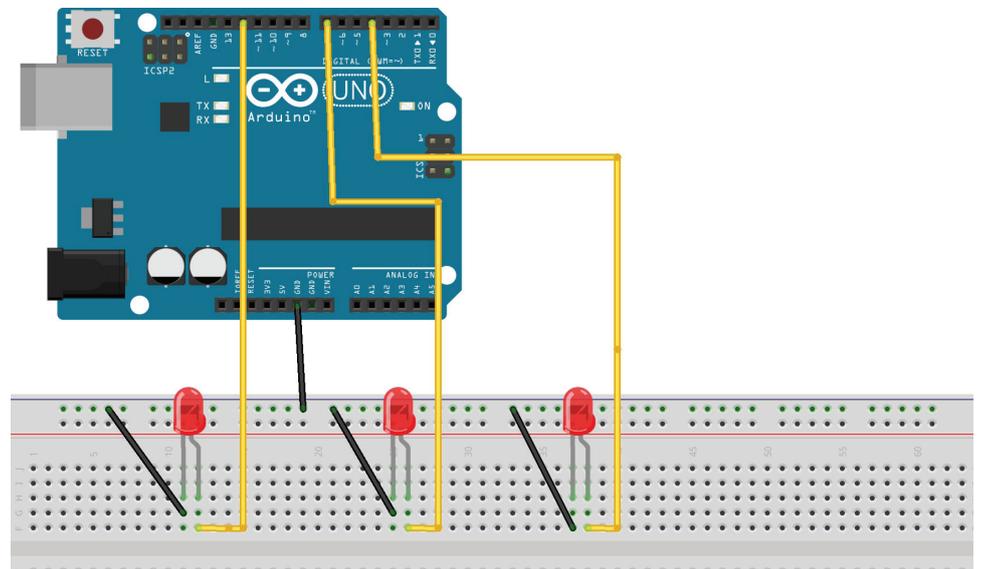




Arduino - Tutorial #3 - Comandi da tastiera

materiale utilizzato – Arduino, cavi elettrici, breadboard, 3 led

In questo tutorial vediamo come accendere e spegnere un Led tramite la tastiera del nostro computer. Questo sketch può risultare utile per tutti coloro che vogliono crearsi una postazione di controllo dal proprio pc per gestire qualsiasi supporto elettronico come luci, motori, pompe d'acqua per annaffiare, elettrodomestici vari, ecc... oppure per gestire video, audio, proiezioni (nel caso di installazioni artistiche o Vj). Dato che siamo all'inizio con le lezioni di Arduino utilizzeremo questo controller keyboard per accendere e spegnere 3 Led. L'utilizzo dei Led serve in questi primi tutorial per spiegare l'informatica e la logica di programmazione di Arduino, in futuro verranno sostituiti con circuiti e componenti più complessi (il codice però resterà sempre lo stesso). Realizziamo il circuito.



Inseriamo i 3 Led nella breadboard e colleghiamo le tre gambette corte al segnale GND. Come visto precedentemente dobbiamo portare il segnale GND di Arduino nella fila esterna della breadboard e da questa creare i tre ponti verso i tre Led. Le gambette lunghe dei Led (segnale positivo) dobbiamo invece collegarle ai Pin digitali 4, 7 e 12.

Passiamo subito alla parte informatica. In questo sketch quando verrà premuto il tasto "a" si accenderà il primo Led, con il tasto "b" il secondo e con il tasto "c" il terzo.

1° fase: dichiarazione delle variabili

Come già visto nel tutorial precedente dichiariamo le tre variabili che associamo ai tre Pin per accendere i tre Led.

```
int ledpin_1 = 4;  
int ledpin_2 = 7;  
int ledpin_3 = 12;
```

ledpin_1, ledpin_2 e ledpin_3 sono tre nomi qualsiasi che associamo rispettivamente ai pin digitali 4, 7 e 12 di Arduino.



Inoltre dobbiamo dichiarare una variabile per leggere i comandi provenienti dalla tastiera:

```
char tastiera=0;
```

Dove la parola "char" identifica una variabile che contiene un carattere. tastiera è il nome che diamo a questa variabile ma può essere chiamata in qualsiasi altro modo (a sinistra il codice della prima fase di programmazione).

```
int ledpin_1 = 4;
int ledpin_2 = 7;
int ledpin_3 = 12;
char tastiera=0;
```

```
void setup(){
pinMode(ledpin_1, OUTPUT);
pinMode(ledpin_2, OUTPUT);
pinMode(ledpin_3, OUTPUT);
Serial.begin(9600);
}
```

2° fase: funzione setup()

Nella funzione setup() dobbiamo dichiarare se i tre Pin utilizzati sono di Output o di Input. In questo caso dato che attraverso i tre Pin controlliamo l'accensione dei Led inviando un segnale dobbiamo dichiararli di Output.

```
void setup() {
pinMode(ledpin_1, OUTPUT);
pinMode(ledpin_2, OUTPUT);
pinMode(ledpin_3, OUTPUT);

}
```

Aggiungiamo all'interno della funzione setup() il comando: `Serial.begin(9600);` Questa istruzione permette ad Arduino di comunicare con il computer tramite il cavo USB. In questo modo digitando le lettere sulla tastiera verranno lette dal nostro programma.

```
void loop(){

tastiera = Serial.read();
switch(tastiera){

case 'a':
digitalWrite(ledpin_1,1);
break;

case 'b':
digitalWrite(ledpin_2,1);
break;

case 'c':
digitalWrite(ledpin_3,1);
break;

default:
Serial.flush();

}

}
```

3° fase: funzione loop()

In questa parte di codice incontriamo il primo controllo sulle variabili. La **switch case** è infatti un'istruzione per gestire più casi a seconda del valore di una variabile. In questo esempio la variabile "tastiera" può assumere diversi valori a seconda dei tasti che digitiamo. Se il tasto premuto è una "a", una "b" o una "c" allora si accenderanno dei Led, altrimenti non succederà nulla.

```
tastiera = Serial.read();
```

legge il comando proveniente dal pc (tramite seriale). Nella variabile "tastiera" viene memorizzato l'ultimo tasto premuto dall'utente.

```
switch (tastiera){
}
```

controlla se il valore di "tastiera" è uguale a un valore predefinito che dichiariamo subito dopo:

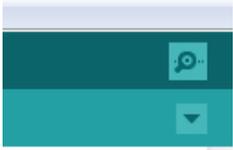
```
case 'a':
digitalWrite(ledpin_1,1);
break;
```

Se "tastiera" è uguale ad "a" (se abbiamo premuto il tasto "a") allora scrivi sul Pin Digitale ledpin_1 il valore 1 cioè accendi il Led. Copiamo nello sketch di Arduino il codice a sinistra

`break;` termina l'operazione.

`default:` se il valore non è presente nei vari controlli allora viene ignorato.

N.B: alla fine del termine default ci sono i due punti e non il punto e virgola.



`Serial.flush()`; è un'altra istruzione che serve per svuotare il buffer dei dati seriali in arrivo dal pc.

I comandi `Serial.begin(9600)`; e `Serial.flush()`; per il momento accettiamoli senza troppi approfondimenti. Dobbiamo solamente sapere che servono per far comunicare Arduino con i dati provenienti dal computer.

Salviamo il nostro sketch nella cartella che abbiamo creato nel primo tutorial e chiamiamolo **2_comando_tastiera**.

Compiliamo lo sketch per verificare eventuali errori (click sull'icona in alto con la "V") e uploadiamo il programma su Arduino (click sull'icona con la freccia rivolta verso destra).

Per far funzionare i comandi della tastiera dobbiamo accendere il monitor seriale di Arduino.

Click sull'icona in alto a sinistra (all'interno c'è una lente di ingrandimento).

In questo modo si apre un piccolo schermo che serve per visualizzare le operazioni e i vari flussi di dati tra Arduino e il computer (più avanti verrà utilizzato per leggere i valori dei sensori collegati ad Arduino). In questo caso dobbiamo cliccare sulla barra vuota in alto, digitare la lettera "a" e premere **Send**.



Se tutto è andato bene il primo Led si accenderà. Digitiamo anche le lettere "b" e "c", sempre premendo Send, per accendere anche il secondo e terzo Led.

Come possiamo fare per accendere un Led e contemporaneamente spegnere gli altri due? Nell'istruzione case oltre al `digitalWrite 1` sul Led scelto dobbiamo scrivere il comando `digitalWrite 0` sugli altri due Led.

Provate a fare questo semplice esercizio senza continuare a leggere qui sotto...

(Nei prossimi tutorial vi inviterò a fare delle piccole modifiche al programma per capire meglio la logica di programmazione. Cercate di eseguire gli esercizi senza copiare le risposte altrimenti è inutile andare. Il copia e incolla di interi sketch di Arduino può essere fatto solamente quando abbiamo preso un pò di esperienza con il codice e sappiamo esattamente cosa stiamo copiando).

Ecco la soluzione:

```
case 'a':  
digitalWrite(ledpin_1,1);  
digitalWrite(ledpin_2,0);  
digitalWrite(ledpin_3,0);  
break;
```

```
case 'b':  
digitalWrite(ledpin_1,0);  
digitalWrite(ledpin_2,1);  
digitalWrite(ledpin_3,0);  
break;
```

```
case 'c':  
digitalWrite(ledpin_1,0);  
digitalWrite(ledpin_2,0);  
digitalWrite(ledpin_3,1);  
break;
```

N.B: se copiate questo codice all'interno dello sketch di Arduino il carattere ' (apostrofo) potrebbe darvi errore. Nel programma di Arduino eliminate l'apostrofo e sostituite con quello della vostra tastiera. Se tutto è andato bene la lettera all'interno dei due ' ' diventa blu.