



Arduino - Tutorial #4 - Lettura dati sensore

materiale utilizzato – Arduino, cavi elettrici, breadboard, pannello solare (o un altro tipo di sensore)

In questo tutorial viene spiegato cos'è un sensore, dove è possibile acquistarlo, come collegarlo e leggere i suoi dati.

Ma a cosa serve un sensore?

Un sensore serve per trasformare eventi che accadono nella realtà in informazioni e segnali che possono essere letti da Arduino. Ad esempio il sensore di alcool è in grado di trasformare la presenza di alcool presente nel nostro corpo in un numero analogico che può variare da 0 a 800 a seconda di quanto abbiamo bevuto. Una volta avvenuta questa trasformazione da valore reale a valore numerico, possiamo utilizzare questo dato per realizzare i nostri progetti. In questo tutorial vediamo come possiamo leggere con Arduino questi dati, nei prossimi tutorial impareremo ad utilizzarli.

Tutti i sensori analogici convertono un'informazione reale in un numero analogico. Esistono moltissimi tipi di sensori, vediamo molto brevemente i più utilizzati:

Distanza: per misurare la distanza tra il sensore e gli oggetti che vengono posti davanti. Molto utilizzati nella robotica per individuare i vari ostacoli che il robot può trovare davanti a sé. Si suddividono in due tipi: infrarossi e ultrasuoni (nei prossimi tutorial utilizzeremo tutti e due i tipi e verrà spiegato il loro funzionamento). I sensori di distanza rilasciano un numero per indicare la distanza con l'oggetto o con la persona che hanno davanti.

Giroscopio o bussola magnetica: è un sensore che rileva l'esatta posizione dell'oggetto. in cui è situato. In questo modo possiamo sapere se l'oggetto che stiamo utilizzando è capovolto, in piedi, su un lato, ecc..

Accelerometro: rileva l'accelerazione che sta subendo l'oggetto.

L'accelerometro e il giroscopio sono ormai presenti in tutti gli smartphone di ultima generazione.

Luce: sensori che rilasciano un determinato numero in base alla luce presente nel luogo. In questo modo possiamo ad esempio sapere se è notte o giorno e di conseguenza attivare uno sketch in Arduino.

I sensori possono essere acquistati online oppure nei negozi di elettronica o modellistica.

Qui sotto alcuni link utili:

<http://www.robot-italy.com> - nella colonna a sinistra sotto la voce "sensori"

<http://www.futurashop.it/> - nella sezione "robotica"

<http://www.arduiner.com/it/> - nella sezione "sensori"

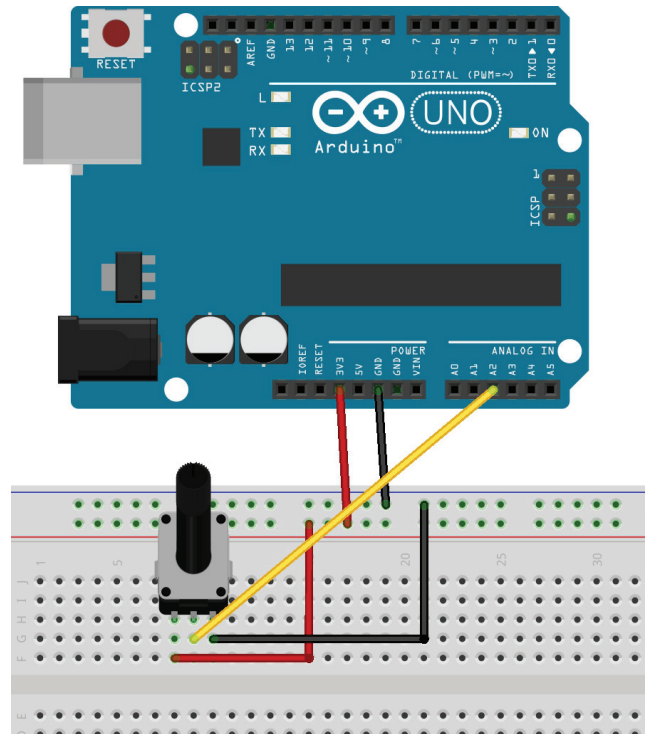
Quando effettuate un acquisto online cercate di prender più componenti possibili per ammortizzare le spese di spedizione (circa 6 euro).



LEGGIAMO I DATI DEI SENSORI

Prima di vedere la parte riguardante il codice da uplodare, realizziamo il circuito collegando un sensore. Non tutti, soprattutto se si è alle prime armi con Arduino, possiedono un sensore. In questo caso vediamo inizialmente un semplice circuito con un potenziometro collegato (simuliamo un sensore) e analizziamo il codice per leggere il risultato. **Più avanti, una volta spiegata anche la parte informatica, verranno mostrati altri esempi su come collegare o costruire un sensore tenendo la parte di programmazione uguale.**

sensore utilizzato:
potenziometro



Collegiamo al Pin 2 analogico l'uscita del nostro potenziometro (il potenziometro è stato descritto nel primo tutorial, [clicca qui](#) per leggerlo). In questo caso lo utilizziamo per simulare un qualsiasi sensore analogico. Infatti muovendo manualmente la rotellina del potenziometro leggeremo il valore dal Pin 2. In realtà la maggior parte dei sensori non ha bisogno di essere toccata, ma funzionano in modo autonomo. Vediamo ora il codice per poter leggere i dati di un qualsiasi sensore.

Apriamo un nuovo sketch di Arduino e salviamolo con il nome: **4_lettura_sensore**.

1° fase: dichiarazione delle variabili

```
int pin2 = 2;  
int lettura_sensore = 0;
```

pin2 è il nome che diamo al pin analogico 2 di Arduino

lettura_sensore è il nome della variabile che utilizziamo per leggere il dato del sensore.



2° fase: funzione setup()

```
void setup()
{
  pinMode(pin2, INPUT);
  Serial.begin(9600);
}
```

dichiariamo che il pin2 di Arduino è di Input perchè su questo Pin arriveranno i dati del sensore.

Serial.begin(9600); - vedi tutorial precedente - [clicca qui](#).

3° fase: funzione loop()

```
void loop()
{
  Serial.flush();
  delay(1000);
  lettura_sensore = analogRead(pin2);
  Serial.println(lettura_sensore);
}
```

Qui incontriamo per la prima volta il comando **analogRead**.

L'analogRead permette di "leggere" i valori che arrivano in un determinato Pin analogico di Arduino.

Nel nostro caso **lettura_sensore = analogRead(pin2)**; vuol dire che la variabile "lettura_sensore" è uguale alla lettura analogica del pin2. Se ad esempio un sensore indica che la quantità di luce presente nella stanza è uguale a 40, questo valore verrà assegnato alla variabile **lettura_sensore**.

Nel nostro caso ruotando il potenziometro la variabile lettura_sensore assumerà un valore tra 0 e 670 circa.

Il comando **Serial.println(lettura_sensore)**; permette di "stampare" questo valore e visualizzarlo sul monitor seriale di Arduino ([precedente tutorial](#)).

delay(1000); questo comando dopo 1 secondo ripete la lettura e la stampa del dato proveniente dal sensore.

Salviamo lo sketch realizzato, uplodiamo il codice su Arduino e apriamo il monitor seriale per visualizzare i dati del sensore (icona in alto a destra).

Qui sotto il codice completo:

```
int pin2= 2;
int lettura_sensore = 0;

void setup(){
  pinMode(pin2, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.flush();
  delay(1000);
  lettura_sensore = analogRead(pin2);
  Serial.println(lettura_sensore);
}
```



Utilizzando lo stesso codice proviamo ora a sostituire qualche sensore.

SENSORE DI LUCE

Per questo secondo esempio utilizziamo un sensore di luce. Al posto del sensore che possiamo acquistare online (fotoresistenza) utilizziamo un pannellino solare (molto più facile da trovare in casa senza dover fare acquisti online). Smontate il pannellino che riuscite a trovare (ad esempio su giocattoli, lampade Ikea, gadget a energia pulita, ecc... se non lo trovate passate al prossimo esempio).

Un pannello solare, se esposto alla luce, rilascia una tensione. Collegando il pannello solare ad Arduino siamo in grado di conoscere la quantità di luce presente nell'ambiente sotto forma di numero. Il pannello solare quindi è un vero e proprio sensore perchè trasforma un elemento naturale in un numero che possiamo elaborare con la nostra scheda. Il codice da caricare è esattamente quello dell'esercizio precedente. L'unica cosa che dobbiamo cambiare è la parte elettronica e in particolar modo l'ingresso del nostro sensore nel Pin 2 che andiamo a leggere e a stampare.

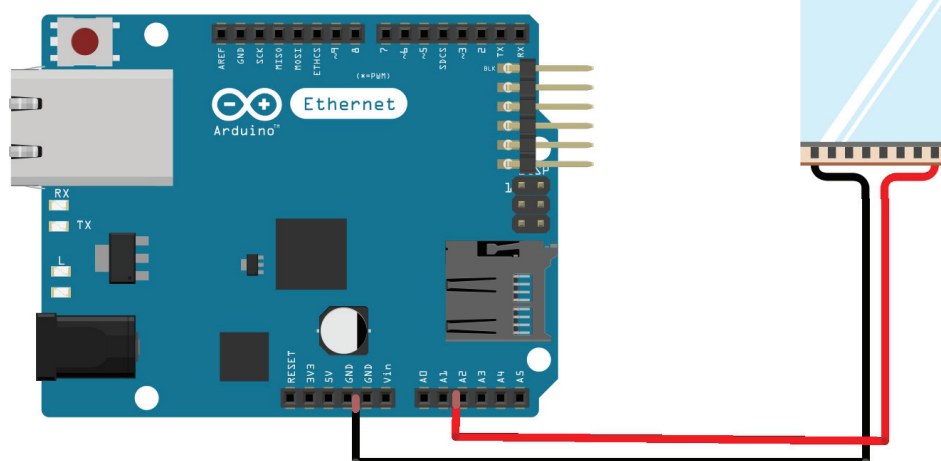
Ogni pannello solare, nel suo retro, ha due connessioni: positivo (+) e negativo (-). Dobbiamo collegare il valore - al Pin GND di arduino e il pin + al nostro Pin2. In questo modo abbiamo chiuso il circuito. Quando il pannello solare viene esposto al sole genera una tensione che viene letta dal Pin2. Se il pannello è al buio la tensione generata sarà uguale a zero.

Colleghiamo il pannello solare (figura sotto) e apriamo il monitor seriale di Arduino. Se spostiamo o copriamo il pannello solare, i numeri in stampa cambieranno da 0 al massimo di tensione generata dal pannello.

Più avanti vedremo come utilizzare questi numeri. Con Arduino è possibile effettuare delle verifiche su questi dati: se il numero che appare sul monitor è uguale a 0 (ovvero è buio) allora Arduino si attiverà per fare quello che gli chiederemo di fare in mancanza di luce, altrimenti se il numero è maggiore di zero, si attiverà un'altra reazione del nostro progetto.

Per il momento continuiamo con la lettura di dati provenienti da altri sensori.

**sensore utilizzato:
pannello solare**



Made with Fritzing.org

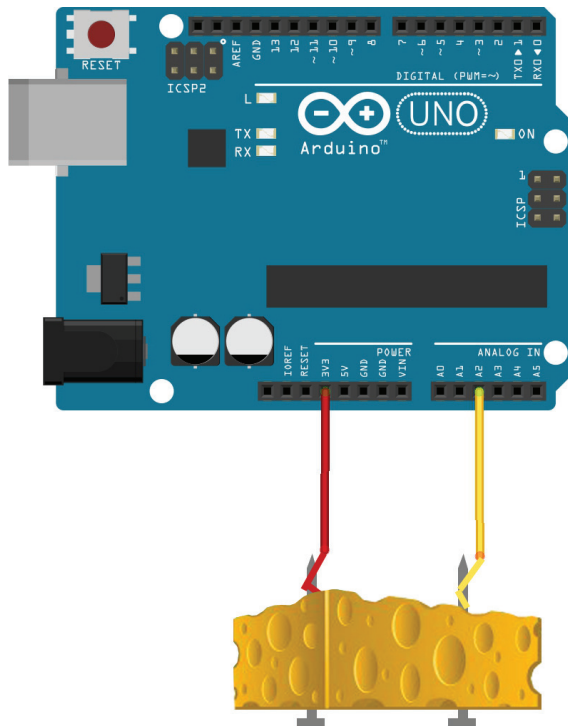


COSTRUIAMO UN SENSORE

Nel prossimo esempio realizziamo un sensore di pressione, ovvero una specie di bottone che, se premuto, restituirà ad Arduino un determinato valore. I sensori di pressione possono essere utilizzati in molte applicazioni, basti pensare a vestiti interattivi (comandare un lettore mp3 semplicemente premendo la tasca della giacca, ecc..), installazioni interattive (un suono parte se una persona si siede su una determinata sedia, ecc..) e molti altri esempi dove un'interattività base con le persone è richiesta.

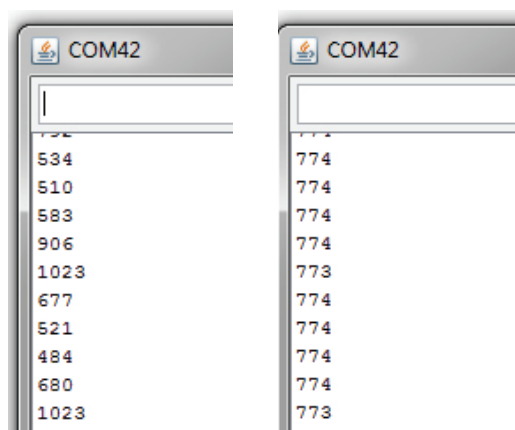
Costruiamo il nostro bottone di gomma piuma: per questo esercizio bastano due chiodini, due cavetti elettrici e un pezzetto di gomma piuma.

sensore utilizzato:
pressione



Inseriamo il primo cavetto elettrico nel Pin2 e l'altra estremità (spellata dalla guaina di plastica) arrotolata ad un chiodino che verrà inserito in un lato della gomma piuma. Il secondo cavetto deve essere inserito nel pin 3V (in modo da inviare una piccola tensione) e la seconda estremità arrotolata ad un altro chiodino anch'esso inserito nel pezzo di gomma piuma.

In questo modo quando la gomma piuma viene premuta ai lati le due "teste" dei chiodini si toccano e il valore di Pin2 diventerà uguale a 774 (il valore numerico di 3V). Qui sotto i due valori presenti nel Serial Monitor di Arduino prima di premere e dopo aver premuto il pezzo di gomma piuma.



Effetto Matrix: il monitor di Arduino e i suoi dati a "cascata" ricordano l'effetto grafico del film Matrix (1999) dove tutti i numeri rappresentano la realtà, l'importante è sapere leggere quei numeri e soprattutto sapere quali andare a prendere.

I sensori trasformano la realtà in numeri e con Arduino siamo in grado di sfruttare questi eventi tradotti in linguaggio informatico.

Dal prossimo tutorial vedremo come utilizzare i dati provenienti dai sensori introducendo i controlli informatici. Queste funzioni non fanno altro che controllare se un numero di un sensore è uguale a un determinato numero che decidiamo noi (valore di soglia). Se questa condizione si verifica allora Arduino compierà un'azione da noi decisa, altrimenti ne eseguirà un'altra.