ISTITUTO TECNICO INDUSTRIALE STATALE
G.GALILEI DI SAN SECONDO

SISTEMI EMBEDDED
Il caso Arduino

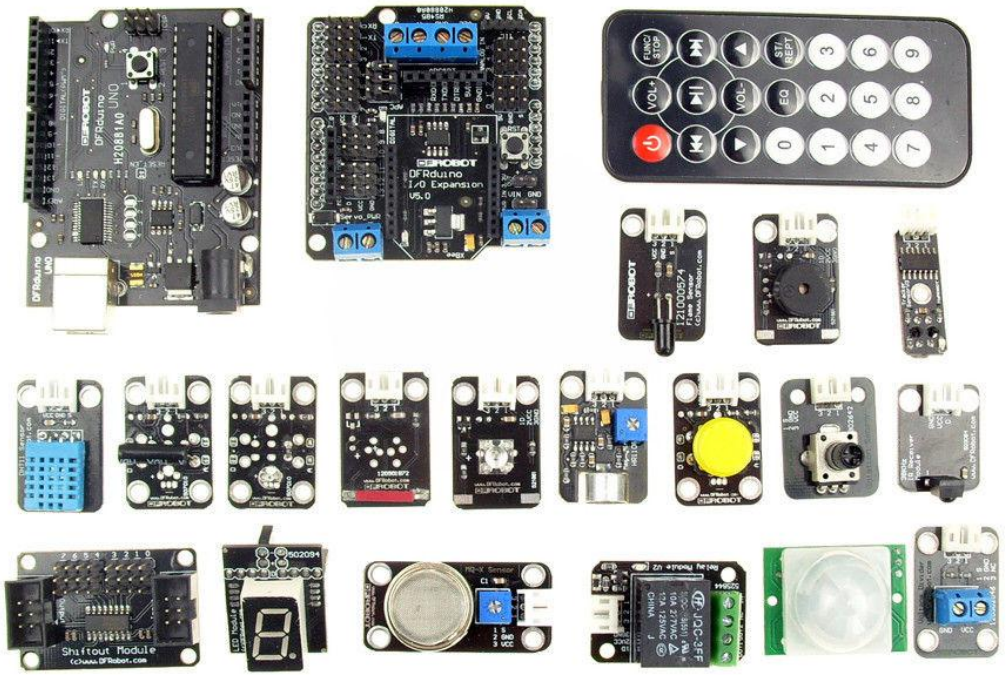
"se ascolto dimentico, se vedo ricordo, se faccio capisco"

INDICE

- I Sensori
- Carichi ad alto assorbimento di corrente

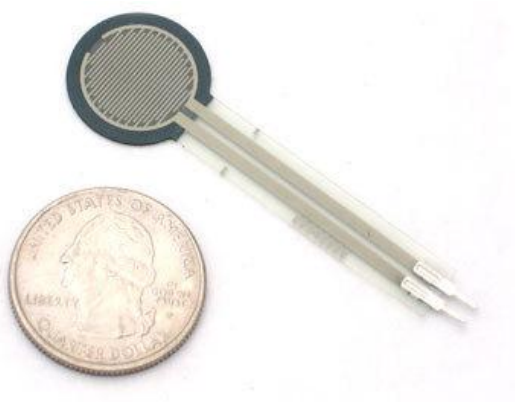
SENSORI

La cos più divertente da fare con Arduino (dopo aver fatto lampeggiare dei LED) è utilizzare dei **sensori per rilevare cosa sta succedendo nel mondo e agire su queste informazioni**. Tuttavia, tutti i sensori hanno i propri metodi di interfacciamento. Il che può essere complicato configurare differenti sensori insieme: alcuni hanno bisogno di resistori pull-up, alcuni devono essere alimentati esternamente, alcuni richiedono una discreta potenza, ecc. Dal momento che esistono molti sensori diversi che le persone tendono a voler usare ho raccolto i sensori più comuni con esempi di codice e schemi di collegamento.



FSR – Forced Sensor Resistor

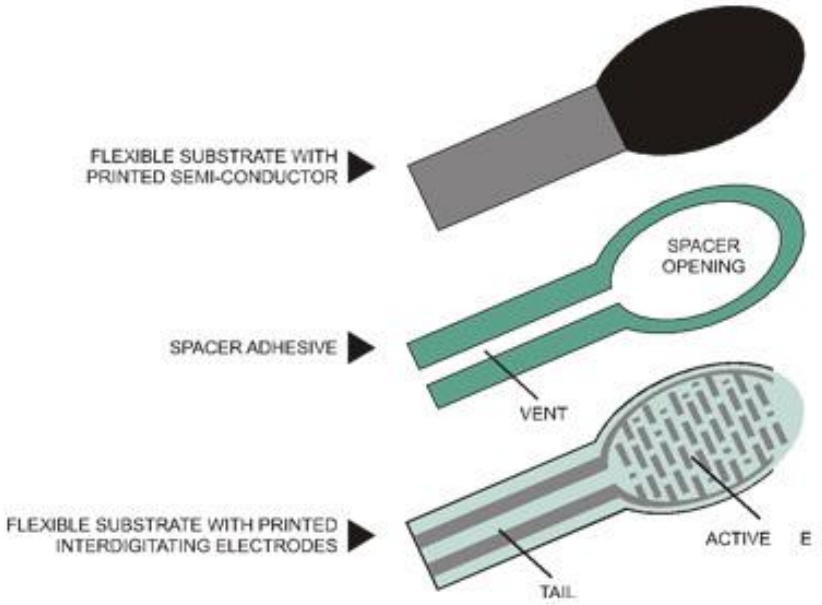
I sensori FSR sono dispositivi che permettono di rilevare la pressione fisica e il peso. Sono semplici da usare e a basso costo. Questa è una foto di un FSR, in particolare il modello **Interlink 402**. La parte sensibile è larga ½'' di diametro.



La FSR è fatto di due strati separati da un distanziatore. Più si preme, più i punti attivi del dispositivo si toccano e il semiconduttore fa scendere la propria resistenza.

FSR – Forced Sensor Resistor

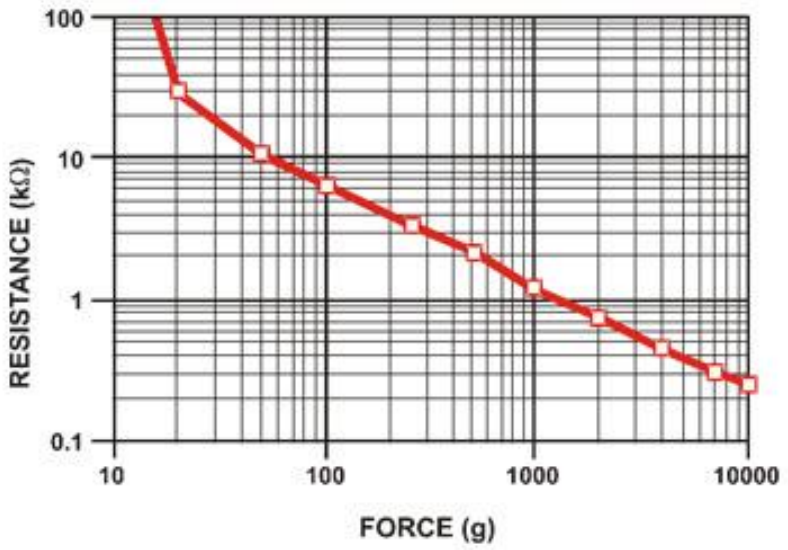
Il sensore FSR è fondamentalmente una **resistenza che cambia il suo valore resistivo** (in ohm Ω) a seconda di quanto sia premuto. Essendo dispositivi a basso costo non riescono a rilevare quanto peso ci sia sopra al dispositivo.
Tuttavia, per la maggior parte delle applicazioni sensibili al tocco come "*ciò è stato schiacciato o spinto e su quanto*" sono ideali per questo sensore.



FSR – Forced Sensor Resistor

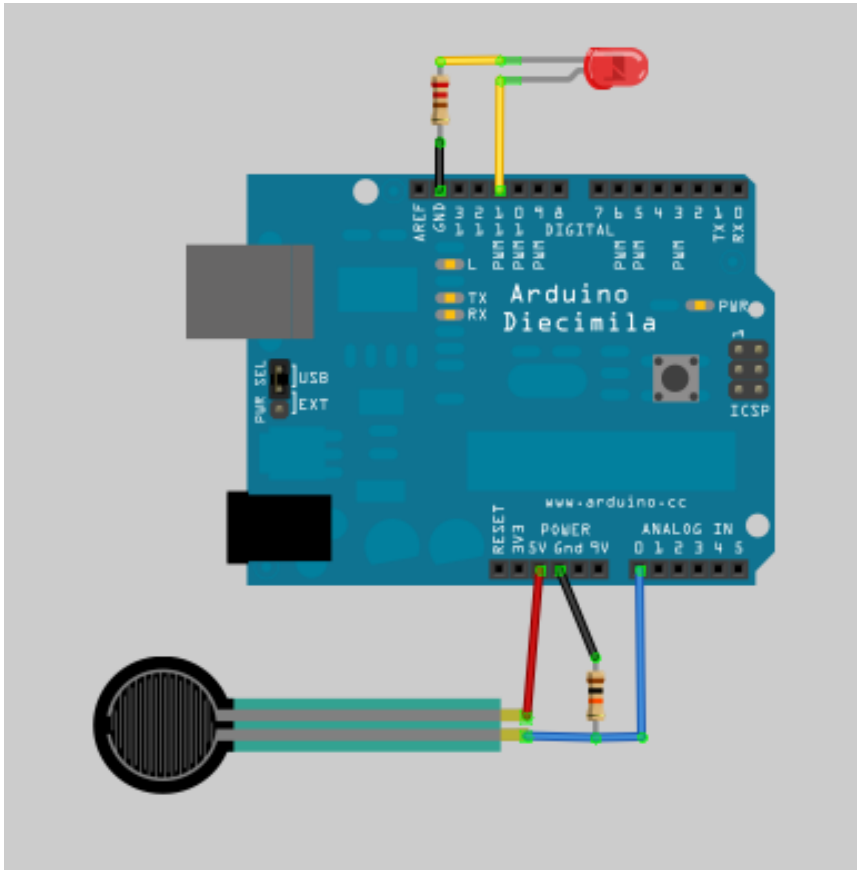
Interlink 402

- Dimensione: 1/2 "(12,5 mm)
- Raggio di resistenza: Circuito aperto (senza pressione), 100KΩ (leggera pressione) a 200Ω (pressione massima)
- Spinta: da 0 a 20 libbre (0-100 Newton) applicata uniformemente sulla 0,125 mq della superficie
- Alimentazione: Qualsiasi! Utilizza meno di 1mA di corrente (dipende dal pullup/resistenze utilizzate e tensione di alimentazione)



FSR – Forced Sensor Resistor

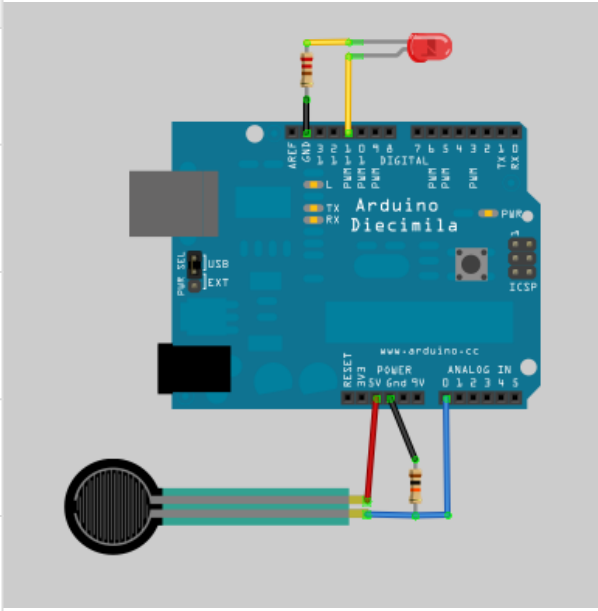
Il modo più semplice per utilizzare questo sensore, è quello di connettere un capo direttamente all'alimentazione, e l'altro metterlo a pull-down con una resistenza da 10KΩ a massa. Il punto di connessione con Arduino (che andrà ad un pin Analogico) lo prenderemo tra la resistenza di PD e il sensore (come lo schema).



Premendo sul sensore, la resistenza dell'FSR diminuisce, e quindi la resistenza totale del FSR e la resistenza del pull-down diminuiscono da circa 100Kohm a 10Kohm. **Ciò significa che la corrente che fluisce attraverso entrambe le resistenze aumenti, e quindi a sua volta provocano un aumento della tensione sulla resistenza da 10K.**

FSR – Forced Sensor Resistor

Force (lb)	Force (N)	FSR Res	(FSR + R)	I on FSR+R	Volt on R
None	None	Infinite	Infinite!	0 mA	0V
0.04 lb	0.2 N	30 Kohm	40 Kohm	0.13 mA	1.3 V
0.22 lb	1 N	6 Kohm	16 Kohm	0.31 mA	3.1 V
2.2 lb	10 N	1 Kohm	11 Kohm	0.45 mA	4.5 V
22 lb	100 N	250 ohm	10.25 Kohm	0.49 mA	4.9 V



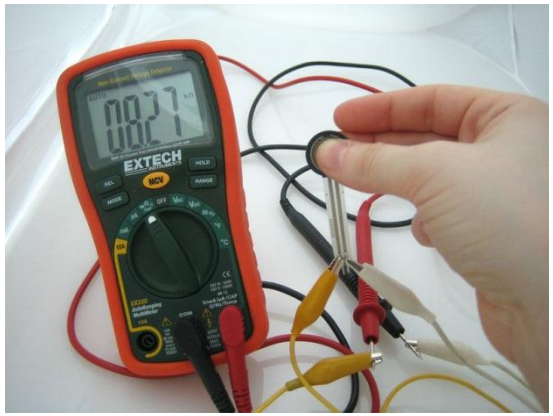
Una tabella che indica approssimativamente i valori di corrente e tensione con un alimentazione di 5v e una resistenza di PD da 10K.

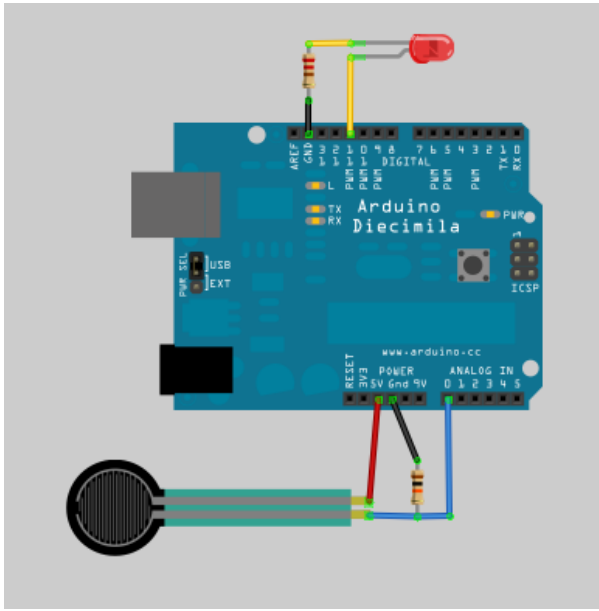
FSR – Forced Sensor Resistor

Vo = Vcc (R / (R + FSR))

Questa è la tensione sulla resistenza ed è inversamente proporzionale alla resistenza dell'FSR.

Il modo più semplice per determinare come il vostro FSR funziona è quello di collegare un tester in modalità resistenza e collegarlo alle due linguette del sensore e vedere come cambia la resistenza.

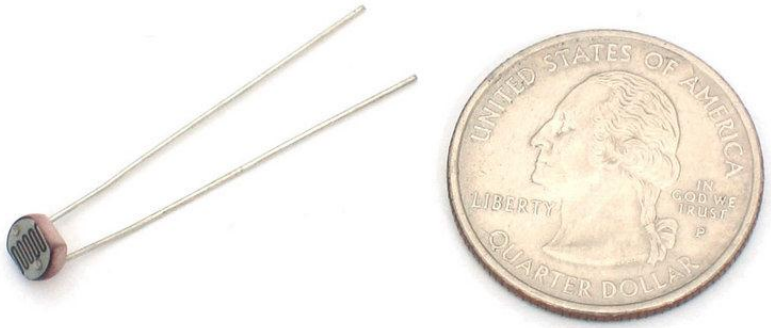




->FSR

Fotoresistenza

Le fotoresistenze sono sensori che permettono di rilevare la luce. **Sono piccoli sensori, poco costosi, di bassa potenza, facile da usare e non si consumano**. Per questo motivo spesso appaiono in giocattoli, gadget e apparecchi elettronici.

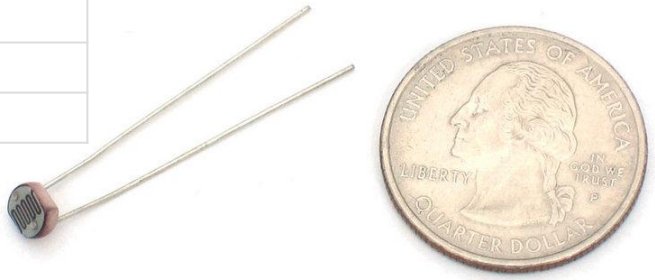


Le fotoresistenze sono fondamentalmente una resistenza che cambia il suo valore resistivo (in ohm Ω) a seconda della quantità di luce che colpisce il sensore. Essendo sensori di bassa qualità ci si può aspettare di essere in grado di determinare cambiamenti considerevoli di luce e non l'esatto valore in lux.

Fotoresistenza

Illuminance	Example
0.002 lux	Moonless clear night sky
0.2 lux	Design minimum for emergency lighting (AS2293).
0.27 - 1 lux	Full moon on a clear night
3.4 lux	Dark limit of civil twilight under a clear sky
50 lux	Family living room
80 lux	Hallway/toilet
100 lux	Very dark overcast day
300 - 500 lux	Sunrise or sunset on a clear day. Well-lit office area.
1,000 lux	Overcast day; typical TV studio lighting
10,000 - 25,000 lux	Full daylight (not direct sun)
32,000 - 130,000 lux	Direct sunlight

Fonte: Wikipedia



Fotoresistenza

PDV-P8001

- Dimensione: circa 5mm di diametro
- Raggio di resistenza: da 200KΩ (buio) a 10KΩ (10 lux di luminosità)
- Sensività: risponde a lunghezze d'onda da 400nm (violetto) a 600nm (arancio).
- Alimentazione: Utilizza meno di 1mA di corrente.

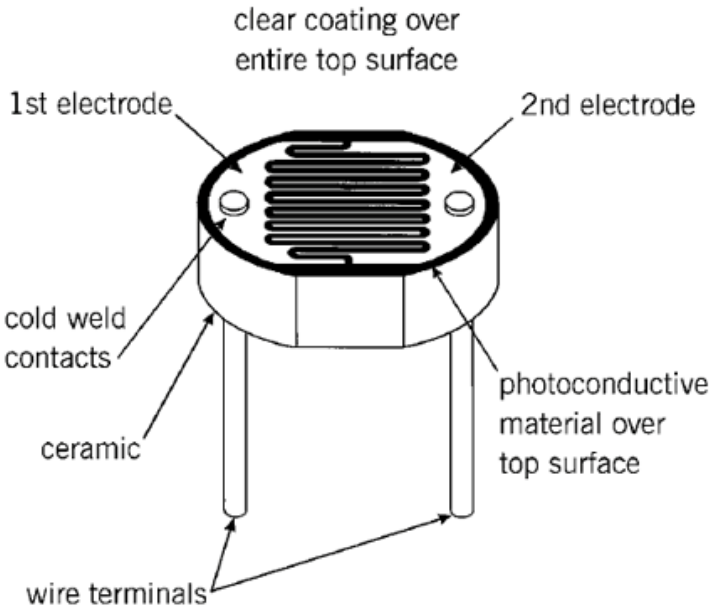


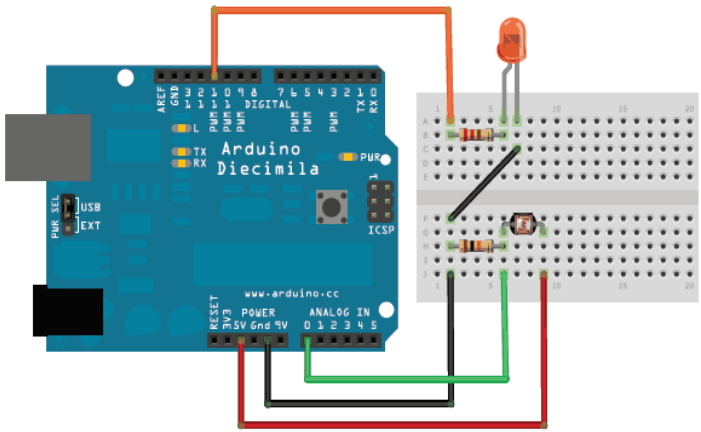
Figure 3
Typical Construction of a Plastic Coated Photocell

Fotoresistenza

Vo = Vcc (R / (R + Fotoresistenza))

Questa è la tensione sulla resistenza ed è inversamente proporzionale alla resistenza dell'FSR.

Il modo più semplice per determinare la resistenza del vostro sensore è utilizzare un tester e misurare la resistenza generata in presenza di luce o meno-

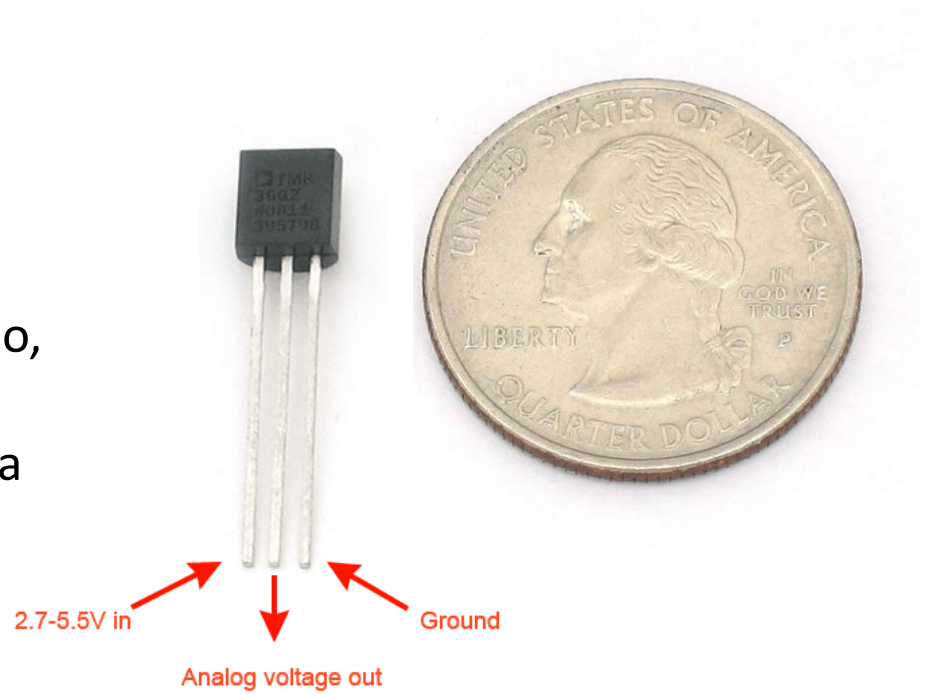


->Fotoresistenza

Sensore di temperatura

Questo sensore permette di misurare la temperatura di un ambiente con una precisione al decimo di grado (0.1 °C).

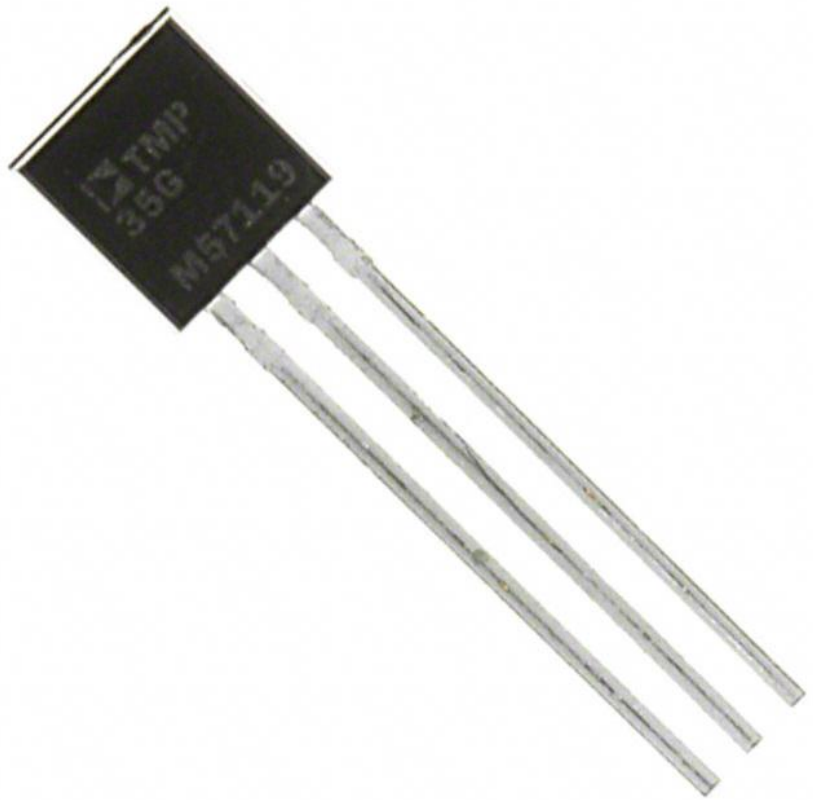
Poiché questi sensori non hanno parti in movimento, sono precisi, non si consumano, non hanno bisogno di taratura, lavorano in diverse condizioni ambientali, ed esiste una notevole consistenza tra il dato letto e rilevato. Inoltre sono molto economici e molto facile da usare.



Sensore di temperatura

TMP35

- Dimensione: circa 0.2’’ x 0.2’’ x 0.2 ‘’
- Range temperatura: da -40°C a 150°C
- Output: 0.1V (-40°C) a 2.0V (150°C).
- Alimentazione: Fino a massimo 5V. Utilizza meno di 1mA di corrente.



Per convertire i volt in temperatura bisogna semplicemente utilizzare questa formula:

Temp in °C = [(Vout in mV) - 500] / 10

Per esempio, se la tensione è di 1V significa che ci sono **((1000 mV - 500) / 10) = 50 °C**

Sensore di temperatura

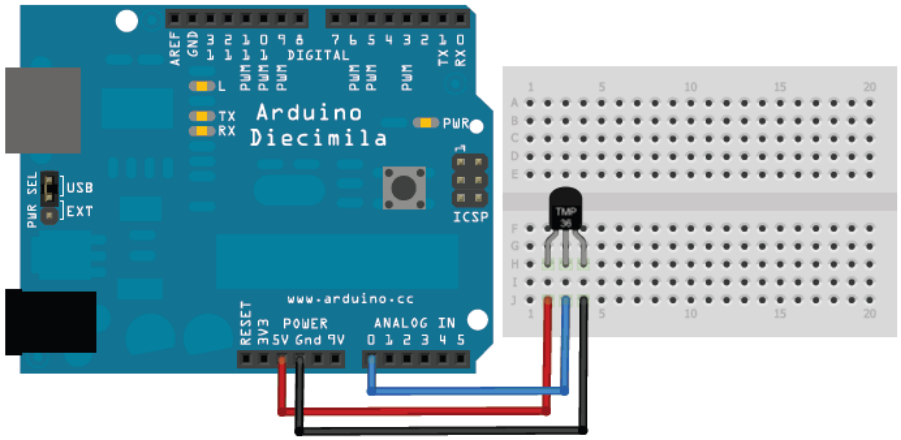
Se si sta utilizzando un Arduino alimentato a 5V è possibile collegare il sensore direttamente ad un ingresso analogico e leggere la tensione sul sensore con la formula:

Voltage at pin in milliVolts = (*reading from ADC*) * (5000/1024)

Per convertire poi la tensione letta in un valore in gradi centigradi devo usare la formula:

Centigrade temperature = [(analog voltage in mV) - 500] / 10

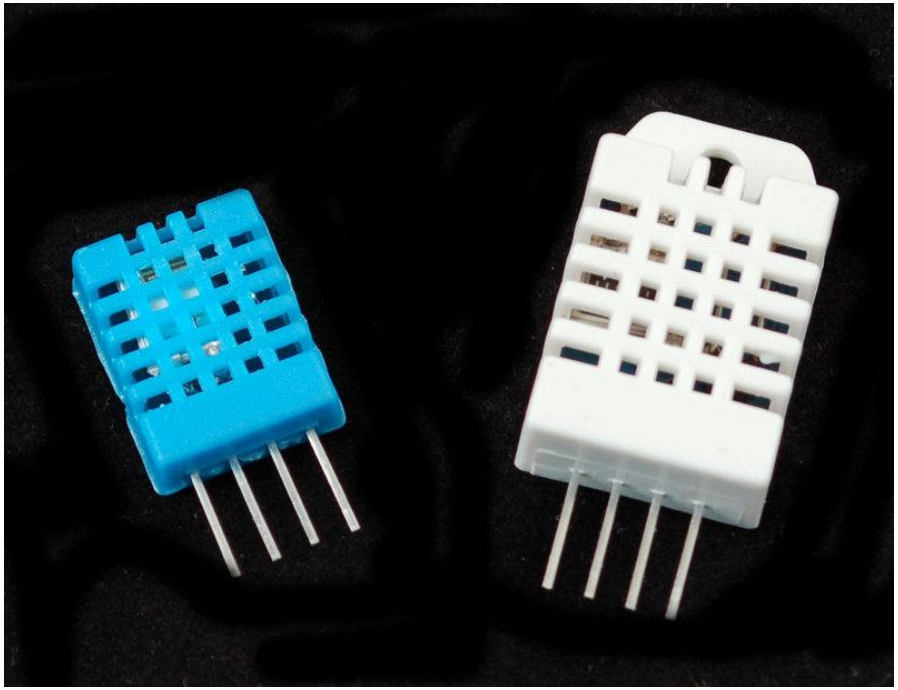
->SensoreTemperatura



Sensore DHT

Questi sensori sono dispositivi a basso costo che rilevano umidità e temperatura. Tali sensori sono molto semplici e lenti, ma sono l'ideale per gli appassionati che vogliono fare un po‘ di data logging di dati ambientali. I sensori DHT sono costituite da due parti: un **sensore capacitivo di umidità e un termistore**.

C'è anche un circuito integrato all'interno che fa una **conversione da analogico a digitale** e restituisce quest’ultimo con la temperatura e l'umidità. Il segnale digitale è abbastanza facile da leggere con qualsiasi microcontrollore.



Sensore DHT

I sensori DHT sono presenti in due versioni

DHT11

- **Ultra low cost**
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate **(once every second)**
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

DHT22

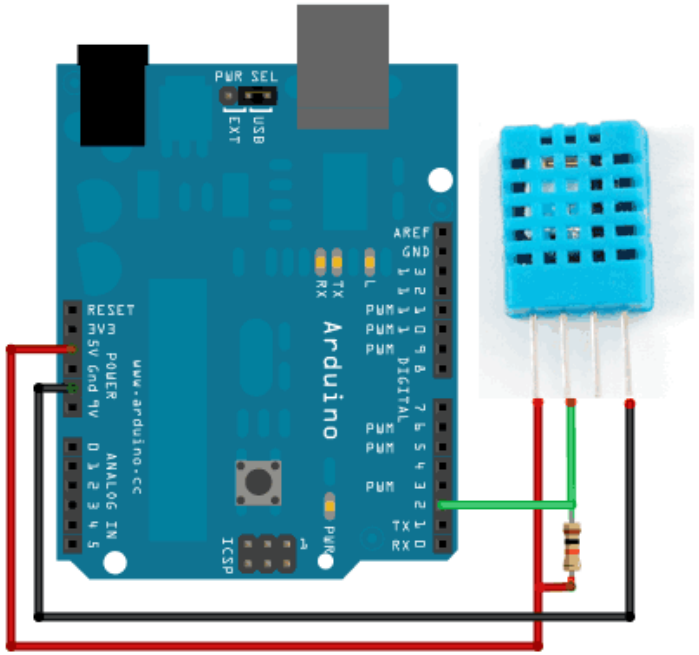
- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 125°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 15.1mm x 25mm x 7.7mm
- 4 pins with 0.1" spacing

Sensore DHT

Il sensore è facile da collegare ad Arduino. Bisogna solo disporre di una resistenza da 10K. I pin di collegamento sono quattro:

- Vcc (da 3V a 5V)
- Data
- NC
- GND

Per utilizzare il sensore devo scaricare la libreria DHT. Essa contiene le funzioni per interfacciarsi con tutte e due le versioni del DHT.

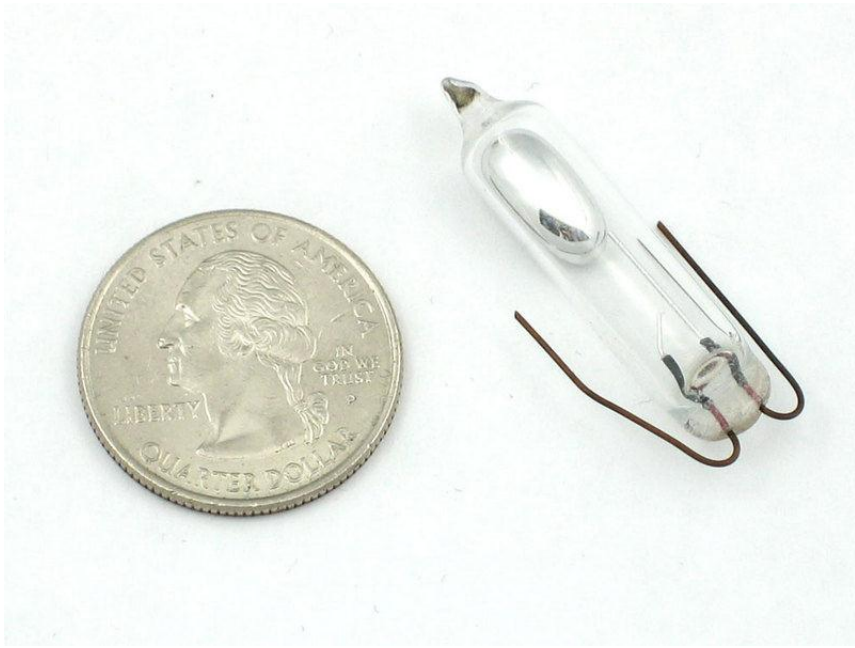


DHT->DHTtester

Sensore di TILT

I sensori di tilt consentono di rilevare **l'orientamento o inclinazione**. Sono piccoli, poco costosi, di bassa potenza e facili da usare. Se usato correttamente, non si usura. La loro semplicità li rende popolare per giocattoli, gadget e apparecchi.

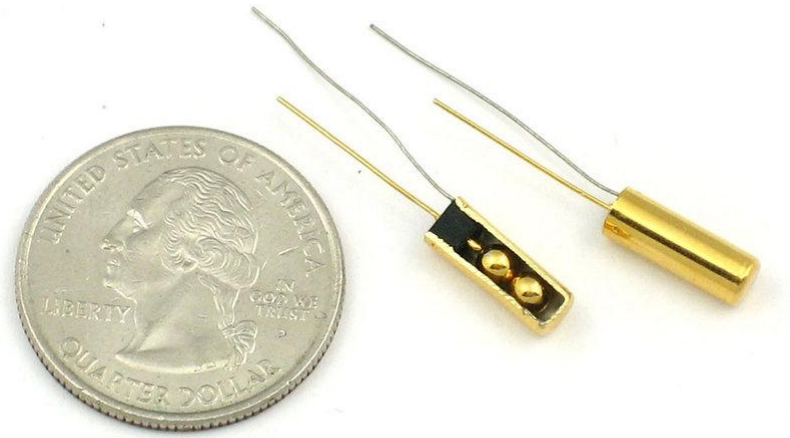
Di solito sono fatti da una cavità di qualche tipo (tipicamente cilindrica, anche se non sempre) e dentro un conduttore di massa libero, come una goccia di mercurio o una pallina che rotola. Una estremità della cavità ha due elementi conduttivi (poli). Quando il sensore è orientato verso il basso, la massa rotola sui conduttori agendo come interruttore.



Sensore di TILT

107-2006-EV
Dimensioni: Cilindrico, 4mm (0.16") diametro 12mm (0.45") lunghezza.
Sensitività: > +-15 gradi
Durata: 50,000+ cicli (switches)
Alimentazione: maggiore di 24V, commuta a 5mA

Anche se non sono così precisi e flessibili come un accelerometro completo, i sensori di tilt sono in grado di rilevare i movimenti più semplici.



->Debounce

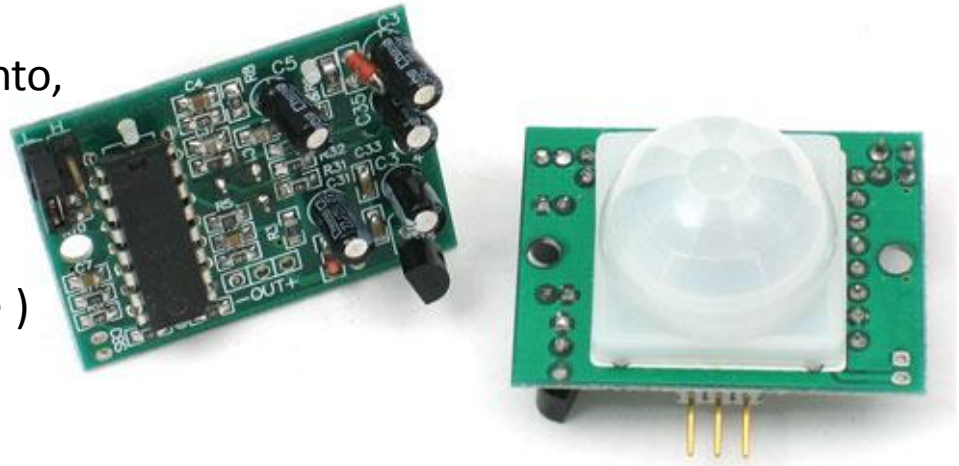
Sensore PIR

Pyroeletric (Passive) InfraRed sensor

Questo sensore, secondo le specifiche rilasciate dal produttore, è in grado di rilevare un oggetto in movimento in una stanza fino ad una distanza di 7 metri e con un angolo di incidenza di 110°. Per alimentarlo basta Arduino con i suoi 5v in quanto il suo **assorbimento è di 50µA**. Il sensore presenta una cupola in plastica, removibile, sfaccettata in modo da permettergli la copertura dei 110° indicati:

Dal sensore fuoriescono 3 cavi di collegamento, da sinistra a destra:

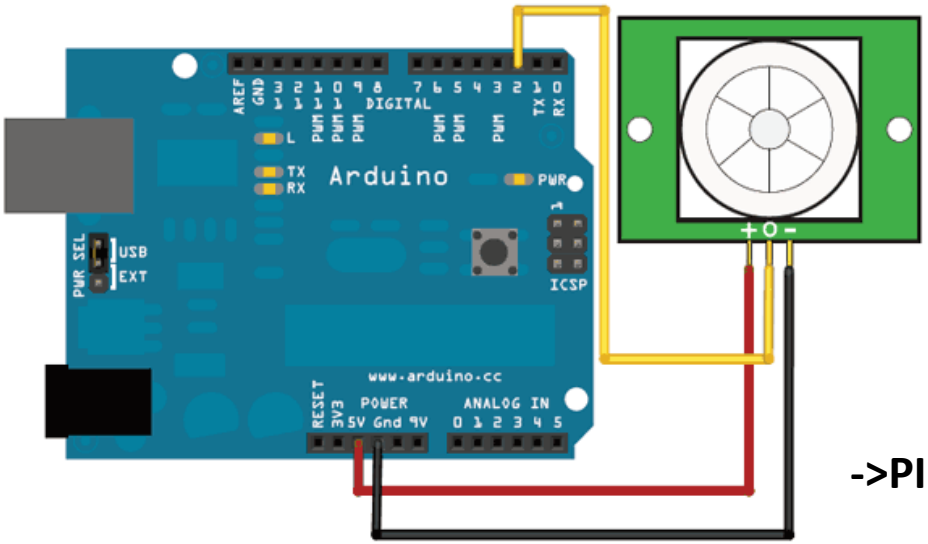
- **nero**: GND o massa (polo negativo di alimentazione)
- **rosso**: 5V (polo positivo di alimentazione)
- **verde**: segnale digitale



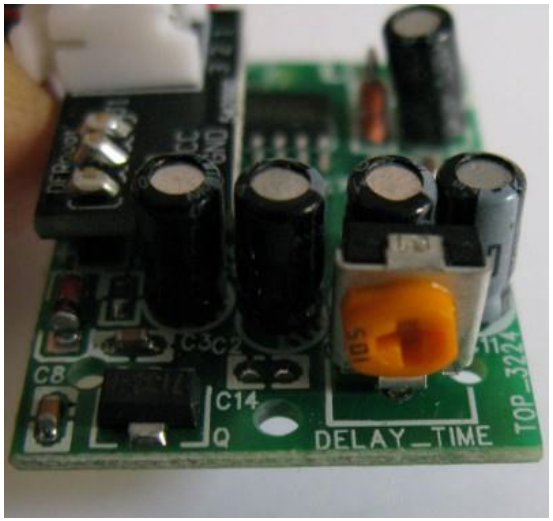
Approfondimento

Sensore PIR

Il funzionamento è semplice: quando rileva un movimento l'uscita del segnale passa da LOW ad HIGH. Secondo le specifiche del costruttore l'uscita passa da 0,4v a 4v, quindi da una soglia inferiore ai 2,5v ad una superiore. Il sensore analizzato presenta anche un potenziometro per regolare il tempo di eccitazione dopo aver rilevato il movimento, ossia il tempo in cui il pin del segnale digitale resta a HIGH prima che, in assenza di movimento, torni a LOW.



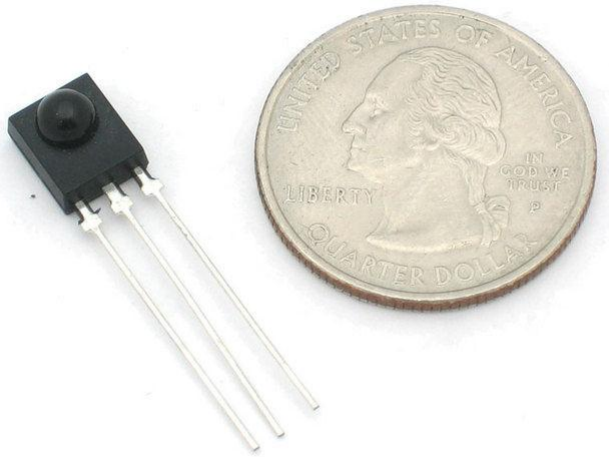
->PIR



Sensore IR

I **sensori IR** sono piccoli microchip con una fotocellula che sono sintonizzati per ‘’leggere’’ **la luce infrarossa**. Essi sono quasi sempre utilizzati per il rilevamento di controllo remoto - tutti i TV e lettore DVD dispongono di uno di questi nella parte anteriore per ascoltare il segnale IR. All'interno del telecomando è presente un corrispondente IR LED, che emette impulsi IR per dire al televisore di attivare, disattivare o cambiare canale.

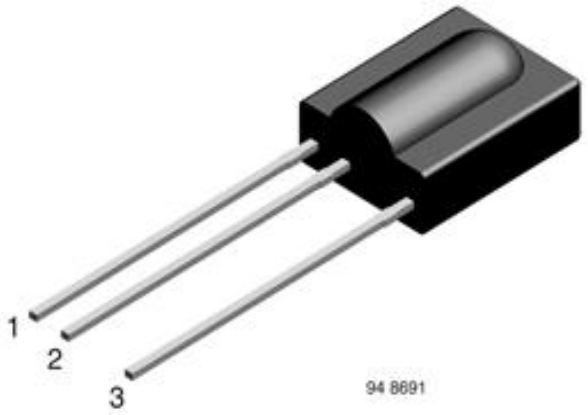
TSOP31238
Dimensioni: quadrato, 7x8mm
Output: 0V (low) se rileva la portante di 38KHz, 5V (high) otherwise
Sensitività: da 800nm a 1100nm (picco a 940nm). Frequenze da 35KHz a 41KHz (picco a 38KHz).
Alimentazione: 3-5V DC 3mA



Sensore IR

Il sensore è possibile collegarlo così:

- Pin 1 GND
- Pin 2 Vs
- Pin 3 Out



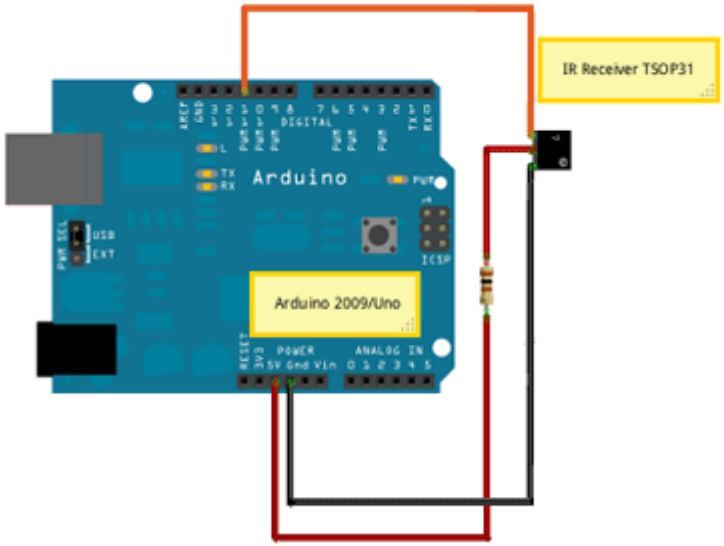
Quando il sensore rileva il segnale IR, mette l'output basso cosicchè può accendere un LED (proprio come accade in una TV).

Lo schema di collegamento è molto semplice, è sufficente collegare una **resistenza da 100Ω tra il terminale positivo e l'alimentazione**.

Sensore IR

Per i test puoi utilizzare la libreria IRremote.h rilasciata sotto licenza creative.

ATTENZIONE: la libreria IRremote.h è veramente fantastica ma presenta un limite, sono decodificati solo alcuni telecomandi (Sony, NEC, RC5, e RC6) se si ha a disposizione un telecomando della Samsung per esempio, qualsiasi segnale inviato dal telecomando sarà visto come 0.



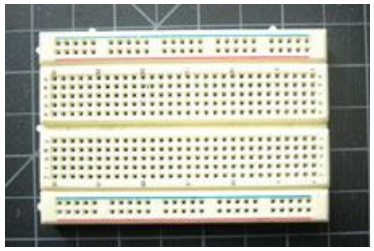
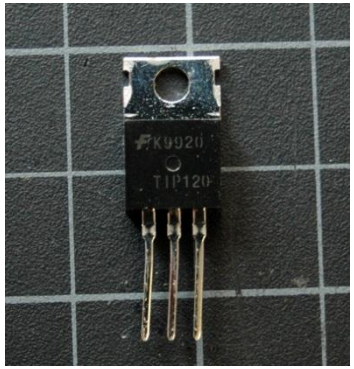
->IRhashDecode

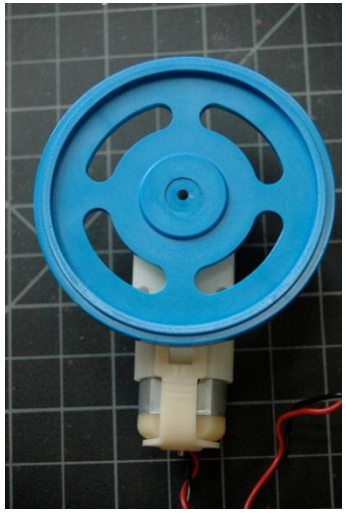
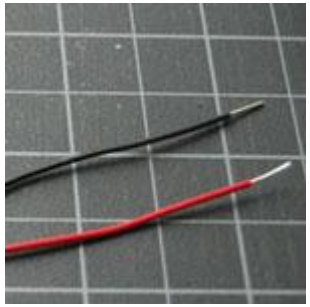
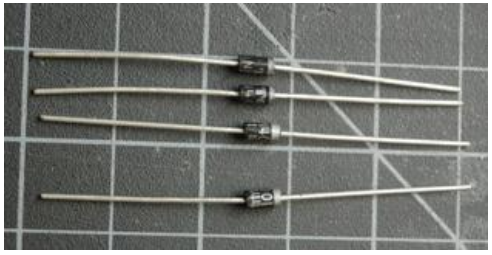
CARICI "High Current"

Controllare un motore DC o una lampadina ad incadenscenza

Occorrente:

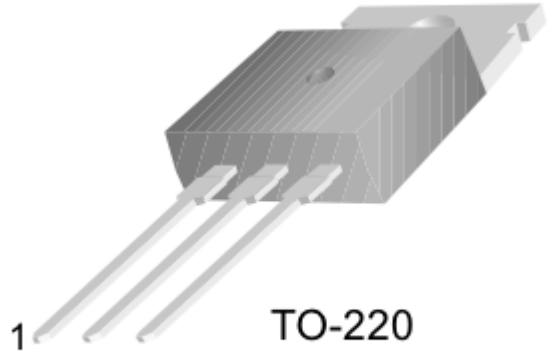
- Breadboard
- Arduino UNO
- Fili
- Potenziometro da 10K
- Diodi
- Transistor TIP120
- Alimentatore esterno
- Motore DC o Lampada incandescenza

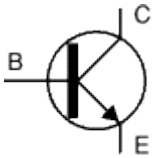




Il transistor

Il transistor consente di controllare un circuito che sta portando maggiore corrente e tensione del microcontrollore. Esso agisce come un **interruttore elettronico**. Quello che si sta utilizzando per questo esempio è un transistor tipo NPN chiamato **TIP120**. E' progettato per la commutazione di carichi ad alta corrente. Dispone di tre connessioni, la base, il collettore e l'emettitore. La base è collegata all'uscita del microcontrollore. L'alta corrente di carico (motore o luce) è collegato alla fonte di alimentazione, e quindi al collettore del transistore. L'emettitore del transistore è collegato a massa.



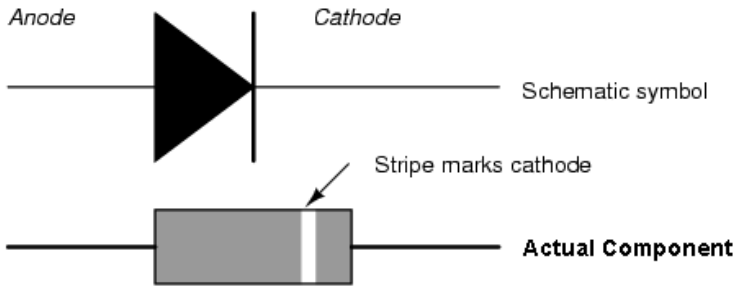
1.Base 2.Collector 3.Emitter



NPN

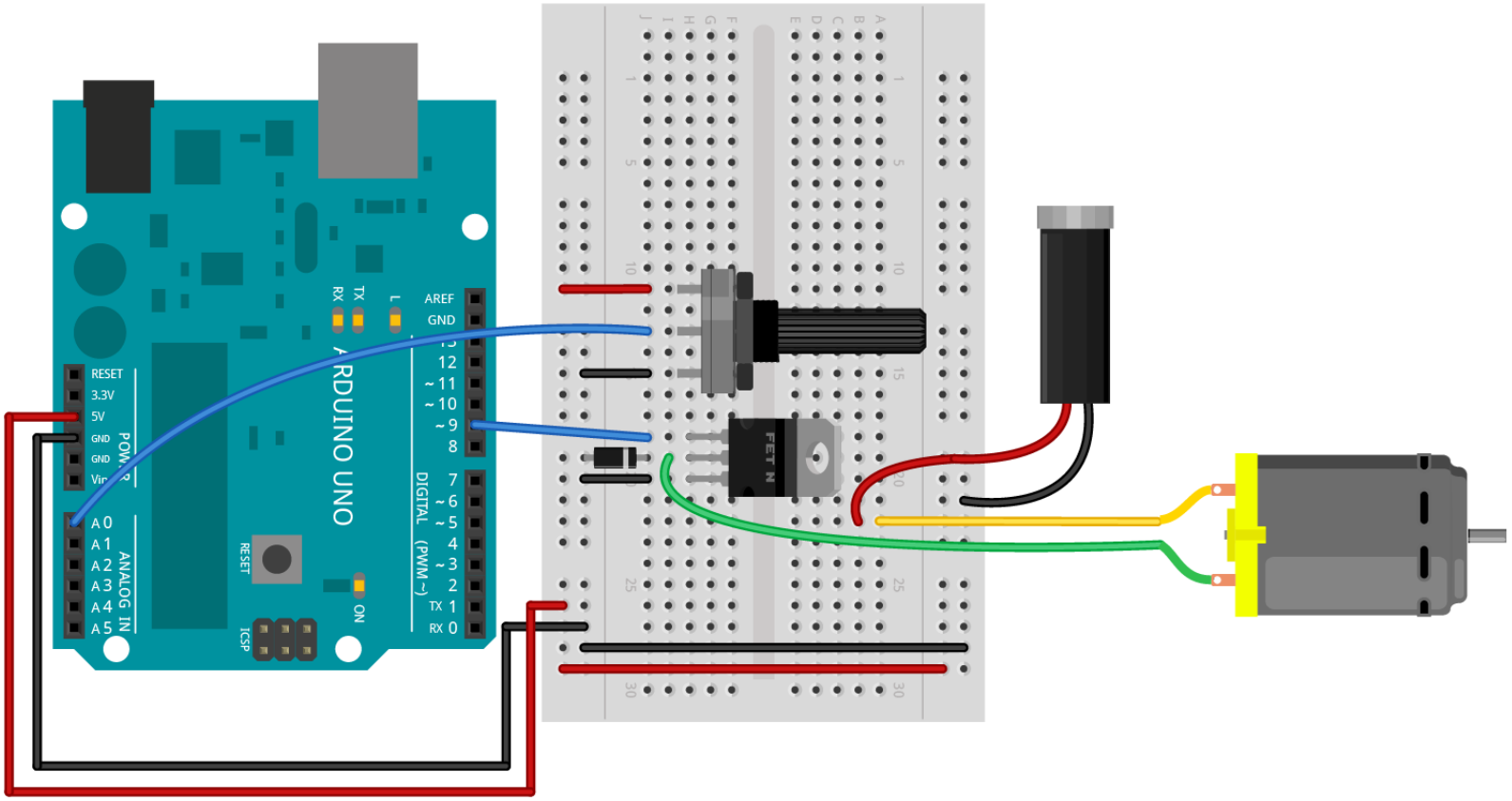
Il diodo

Ora bisogna attaccare il motore in CC al collettore del transistore. La maggior parte dei motori richiederà più amperaggio rispetto a quello che il microcontrollore è in grado di fornire, quindi sarà necessario aggiungere un alimentatore separato. Se il motore gira intorno a 9V, è possibile utilizzare una batteria da 9V. Un motore 5V potrebbe funzionare con 4 batterie AA. **La massa d'alimentazione del motore deve essere collegata alla terra del microcontrollore, sulla breadboard**.

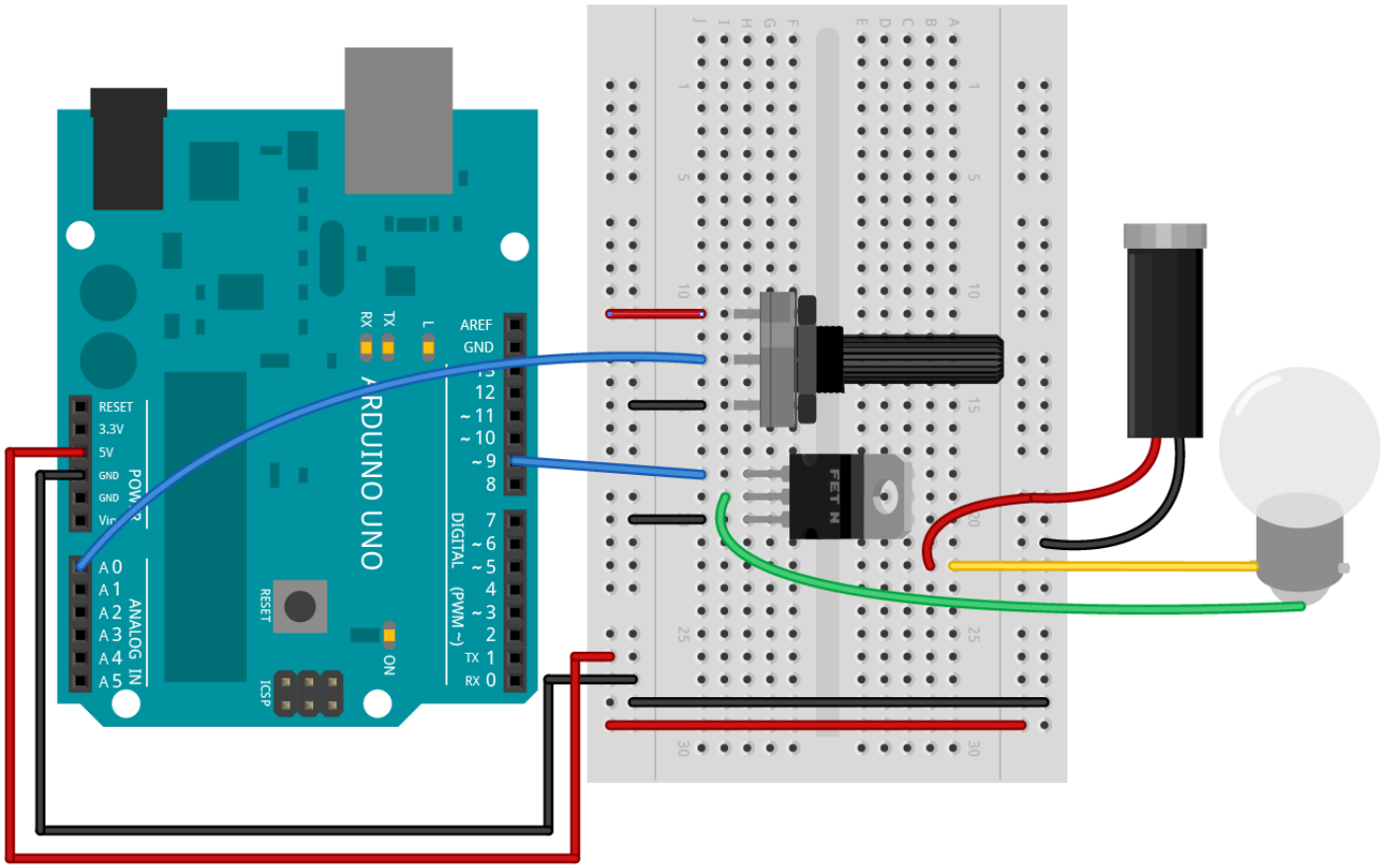


Infine, aggiungere un diodo in parallelo con il collettore e l'emettitore del transistore. Il diodo protegge il transistor dalla tensione di ritorno, generata quando il motore si spegne, o se il motore viene fatto ruotare nella direzione inversa.

Un motore DC


ARDUINO UNO
RESET
3.3V
5V
GND
GND
AREF
GND
ANALOG IN
DIGITAL (PWM~)
FET N

Una lampada ad incandescenza



CARICI "High Current" – Esempio

```
const int transistorPin = 9;    // connected to the base of the transistor

 void setup() {
   // set  the transistor pin as output:
   pinMode(transistorPin, OUTPUT);
 }

 void loop() {
   // read the potentiometer:
   int sensorValue = analogRead(A0);
   // map the sensor value to a range from 0 - 255:
   int outputValue = map(sensorValue, 0, 1023, 0, 255);
   // use that to control the transistor:
   analogWrite(transistorPin, outputValue);
}
```

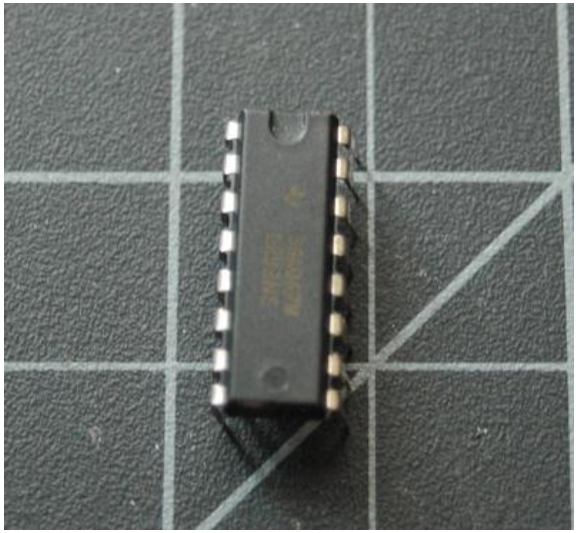
->HIGHCurrentLoads

Il ponte H

Questo esempio si basa sull'L293NE o l'SN754410 della Texas Instrument.

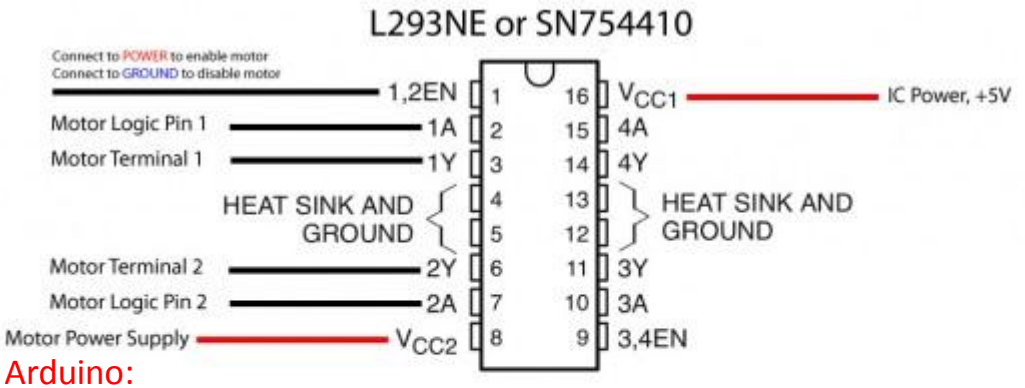
Come funziona un ponte H?

L'L293NE/SN754410 é un ponte-H molto semplice. E' costituito da due "ponti", uno sul lato sinistro del chip ed uno sul lato destro, e può controllare 2 motori. Può controllare fino ad 1 ampere di corrente, e funziona tra i 4.5 V e i 36 V.



Il ponte H

- Pin 1 (1,2EN) attiva/disattiva il nostro motore a seconda che sia HIGH o LOW
- Pin 2 (1A) é un pin logico per il nostro motore (può essere HIGH o LOW)
- Pin 3 (1Y) é uno dei due punti a cui metteremo il motore (motor terminals)
- Pin 4-5 sono per la massa (GND)
- Pin 6 (2Y) é per l'altro motor terminal
- Pin 7 (2A) é un pin logico per il nostro motore (può essere HIGH o LOW)
- Pin 8 (VCC2) é l'alimentazione per il nostro motore, a cui dovremmo dare il voltaggio giusto per il nostro motore
- Pin 9-11 utilizzati per l'altro motore
- Pin 12-13 sono per la massa (GND)
- Pin 14-15 non sono connessi
- Pin 16 (VCC1) é connesso a 5V

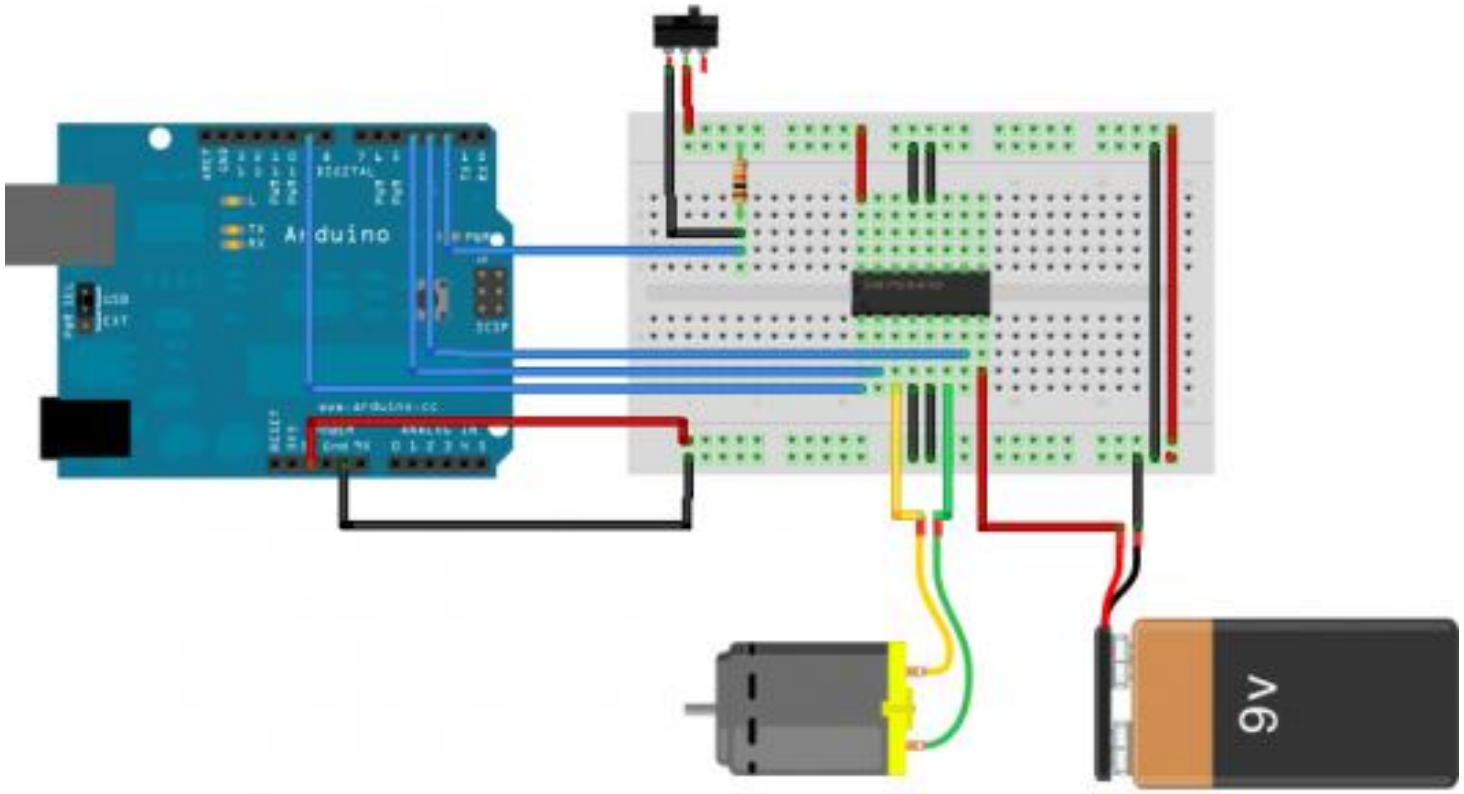


L'EN pin del chip è controllato con un pin digitale di Arduino: settandolo HIGH o LOW accenderemo o spegneremo il motore. Anche i pin logici del chip saranno connessi a dei pin digitali di Arduino, in modo da andare in una direzione con HIGH e LOW, nella direzione opposta con LOW e HIGH. L'alimentazione del motore é comunemente esterna.

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

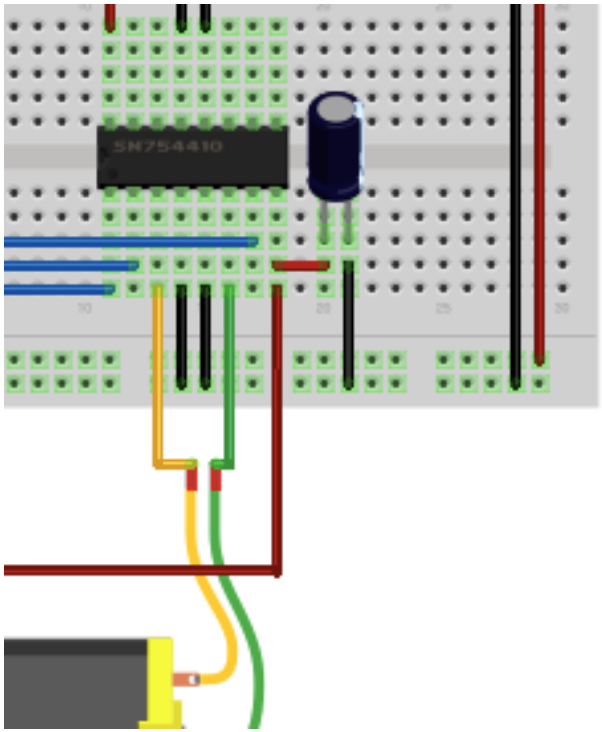
L = low, H = high, X = don't care

Il ponte H



Il ponte H

Se notate che il vostro microcontrollore va in reset quando il motore si accende, aggiungete un condensatore tra alimentazione e massa vicino al motore. Il condensatore spianerà lo sbalzo elettrico che avviene quando il motore si accende.



*This use of a capacitor is called a **decoupling capacitor**. Usually a 10 - 100uF capacitor will work. The larger the cap, the more charge it can hold, but the longer it will take to release its charge*

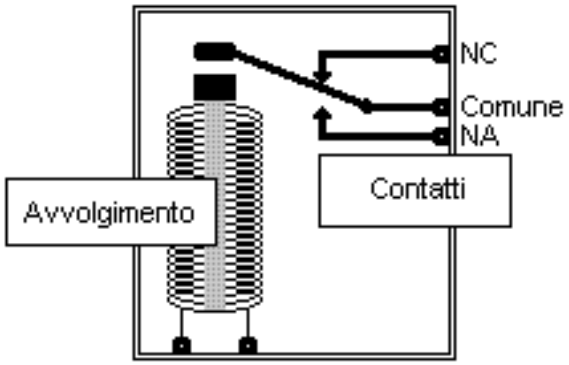
Esempio con ponte H

```
const int switchPin = 2;    // switch input
const int motor1Pin = 3;    // H-bridge leg 1 (pin 2, 1A)
const int motor2Pin = 4;    // H-bridge leg 2 (pin 7, 2A)
const int enablePin = 9;    // H-bridge enable pin
const int ledPin = 13;      // LED

void setup() {
  // set the switch as an input:
  pinMode(switchPin, INPUT);

  // set all the other pins you're using as outputs:
  pinMode(motor1Pin, OUTPUT);
  pinMode(motor2Pin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(ledPin, OUTPUT);

  // set enablePin high so that motor can turn on:
  digitalWrite(enablePin, HIGH);

  // blink the LED 3 times. This should happen only once.
  // if you see the LED blink three times, it means that the
          module
  // reset itself,. probably because the motor caused a
          brownout
  // or a short.
  blink(ledPin, 3, 100);
}
```

```
void loop() {
    // if the switch is high, motor will turn on one
            direction:
    if (digitalRead(switchPin) == HIGH) {
      digitalWrite(motor1Pin, LOW);
      digitalWrite(motor2Pin, HIGH);
    }
    // if the switch is low, motor will turn in the other
            direction:
    else {
      digitalWrite(motor1Pin, HIGH);
      digitalWrite(motor2Pin, LOW);
     }
  }

  /*
    blinks an LED
   */
  void blink(int whatPin, int howManyTimes, int milliSecs) {
    int i = 0;
    for ( i = 0; i < howManyTimes; i++) {
      digitalWrite(whatPin, HIGH);
      delay(milliSecs/2);
      digitalWrite(whatPin, LOW);
      delay(milliSecs/2);
    }
  }
```

I relè

Il relè è un dispositivo elettromeccanico costituito da un avvolgimento e da uno o più contatti meccanici, è utilizzato per operazione di interruzione e commutazione di circuiti elettrici.
Normalmente **viene comandato da un segnale a bassa tensione e i suoi contatti meccanici sono collegati a circuiti di potenza o di utilizzazione**.



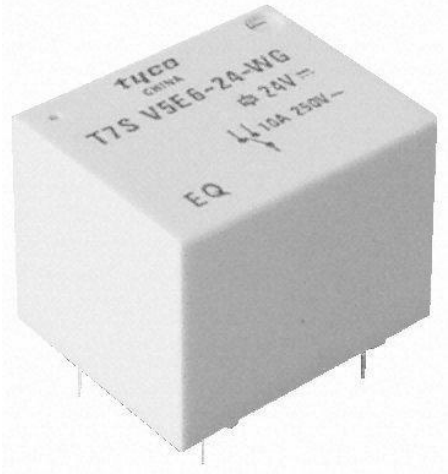
Ne esistono di varie tipologie tra cui

- **Relè Normali**: la commutazione dei contatti avviene quando la bobina è costantemente alimentata.
- **Relè ad impulsi**: la commutazione dei contatti avviene alimentando la bobina per un breve periodo di tempo.
- **Relè a tempo**: la commutazione dei contatti avviene in modo ritardato rispetto al tempo di alimentazione della bobina stessa.

I relè

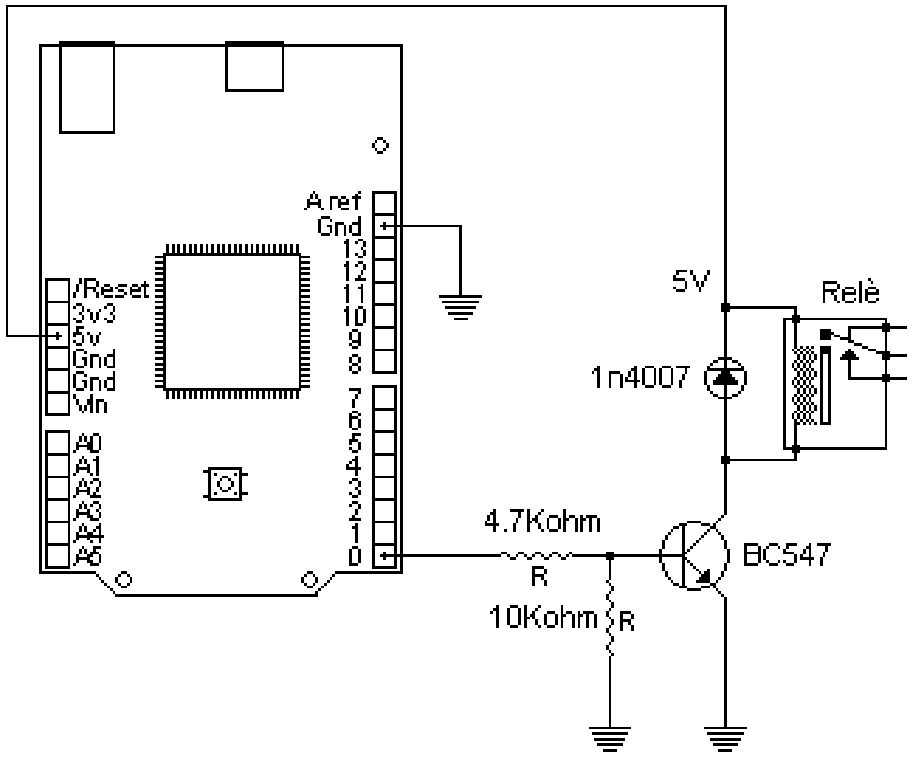
Un relè appropriato può essere il modello è il T7SS5E6-05 prodotto da TycoElectronics.
Questo **attuatore** ha una dimensione molto contenuta e i suoi contatti possono sopportare carichi fino a 250v 6A.
La bobina viene eccitata a 5V è consuma 70mA.
Con Arduino non possiamo alimentare direttamente la bobina, perché le uscite digitali non arrivano ad erogare 70mA ma solamente 50mA. E' necessario quindi, un circuito pilota che permette di interfacciarsi al relè.

Questo solitamente è costituito da un transistor e una coppia di resistenze.
Il circuito seguente mostra quanto esposto:



I relè

L'uscita digitale dell'Arduino è collegata tramite le due resistenze alla base del transistor, l'emettitore è collegato alla massa, il collettore è collegato ad un pin della bobina e l'altro pin della bobina è collegato alla tensione di 5Vdc.
Il **diodo 1n4007**, collegato come in figura, viene impiegato per eliminare le sovratensioni generate dalla bobina.
Per utilizzare un altro tipo di relè, dobbiamo verificare la tensione di alimentazione e la corrente di lavoro della bobina.



Lo schema di principio del circuito pilota rimane invariato, bisogna verificare solamente se la corrente necessaria alla bobina è sopportata anche dal transistor.

Ad esempio se volessi alimentare lo stesso relè con una tensione maggiore, tipo 12Vdc, devo assicurarmi che la bobina sia sempre alimentata con 5Vdc e 70mA.

I relè

Procediamo in questo modo, per prima cosa calcolo la **resistenza della bobina**:

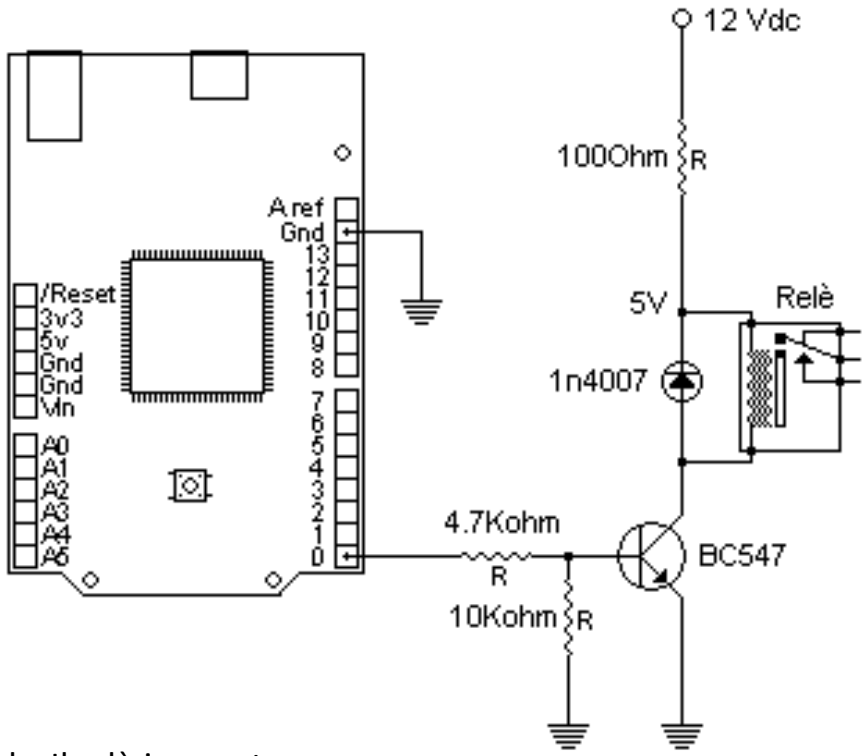
5 Vdc / 70 mA = 71 Ohm

ora calcolo la **resistenza necessaria a far scorrere 70 mA nella bobina con una tensione di 12Vdc**

12 Vdc / 70 mA = 171 Ohm

sapendo che la bobina ha una resistenza di 71 Ohm, calcolo la **resistenza aggiuntiva** da porre in serie alla bobina

171 Ohm – 71 Ohm = 100 Ohm

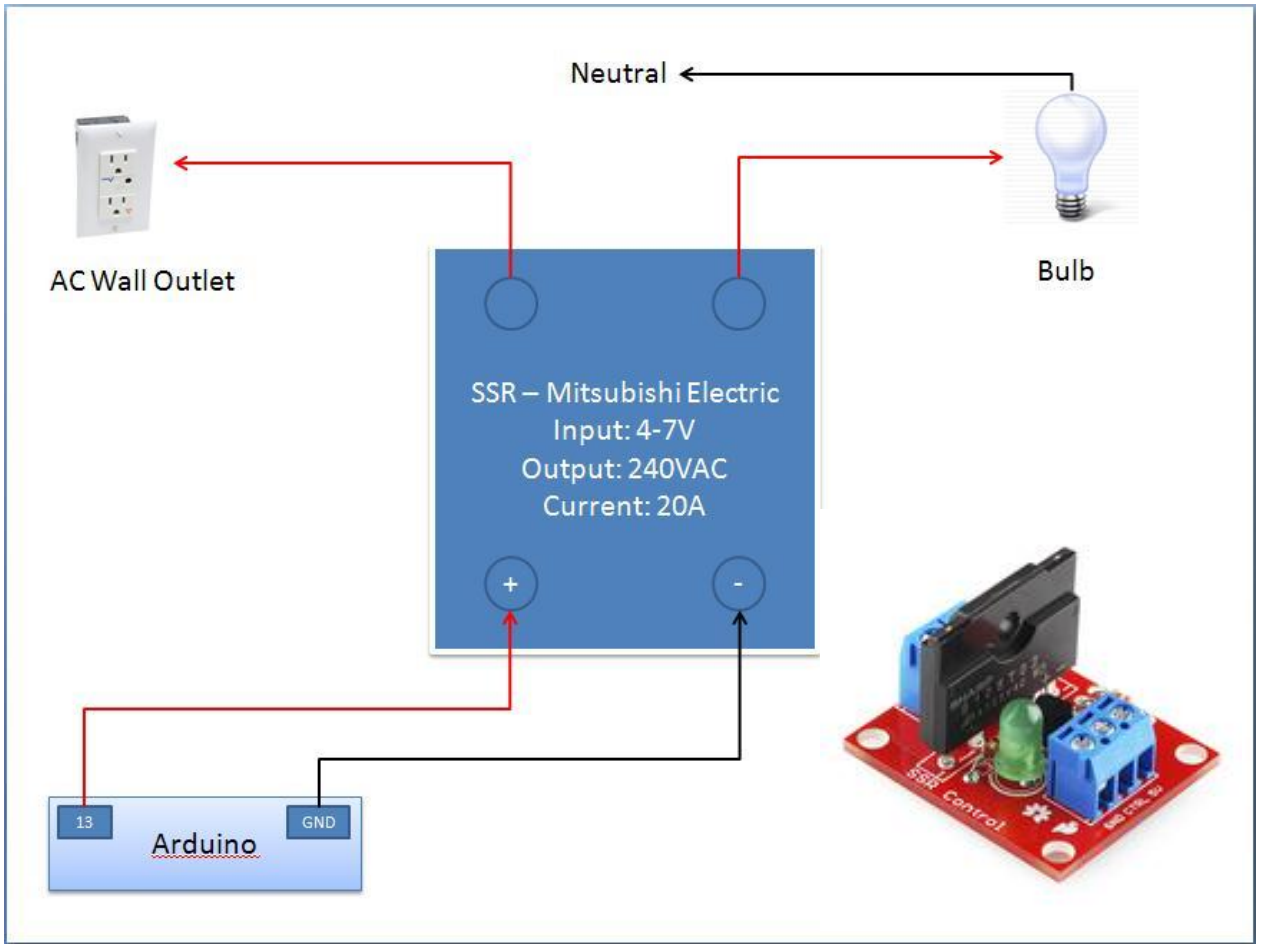


La resistenza di 100 Ohm provoca una caduta di tensione di 7Vdc. Il relè in questo modo è sempre alimentato con una tensione di 5V e 70mA.

I relè - semplificazione

In alternativa si possono usare dei **SSR ovvero Solid State Relais relais** che non hanno parti meccaniche ma hanno all'interno tutta l'elettronica che serve per comandare un carico a 220v controllandolo con un pin di arduino.

Link



RIFERIMENTI

www.mancio90.it

mirkomancin90@gmail.com