

Progetti reali con ARDUINO

Introduzione alla scheda Arduino (parte 5^a)

giugno 2013 – Giorgio Carpignano

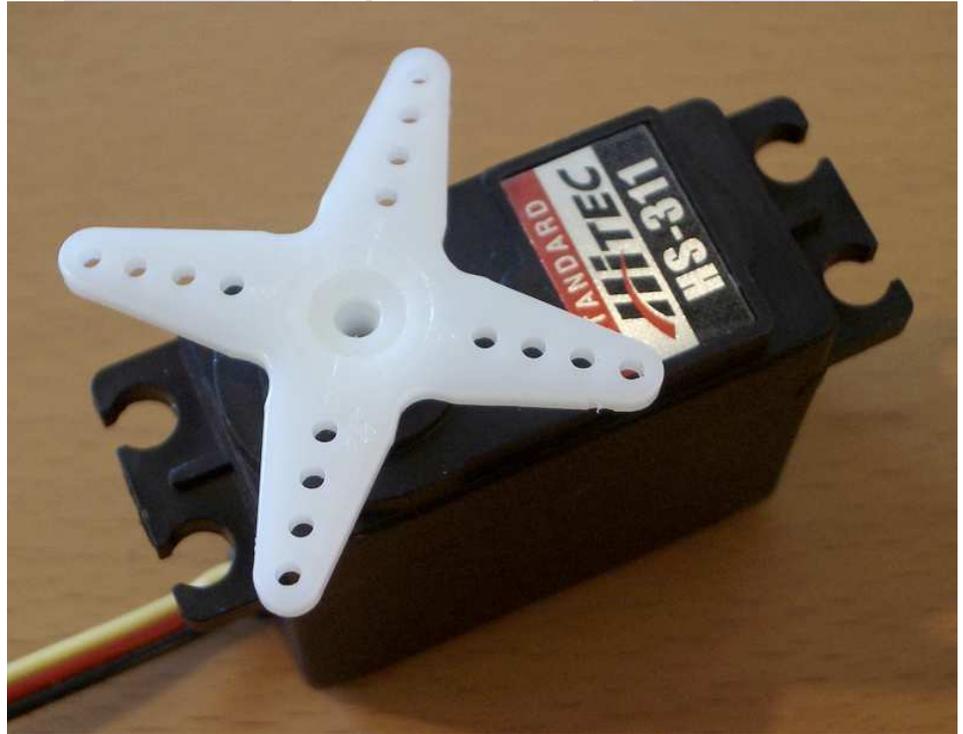
I.I.S. PRIMO LEVI

C.so Unione Sovietica 490 (TO)

Materiale didattico: www.iisprimolevi.it

Servomotori per radiocomandi

- Può essere posizionato con una rotazione dell'albero compreso tra **0°** e **180°**
- Un circuito di feedback interno (controreazione) e ingranaggi si prende cura di mantenere la posizione corretta
- Facile da utilizzare e collegare: utilizza solo 3 fili di collegamento con alimentazione a 5V.
- In generale, il "servomotore" è un motore con un intrinseco meccanismo di **feedback** (composto da sensore di posizionamento e sua logica di controllo) che permette di inviare comandi di posizione allo stesso senza che sia necessario effettuare la lettura di posizione per verificare la corretta posizione.



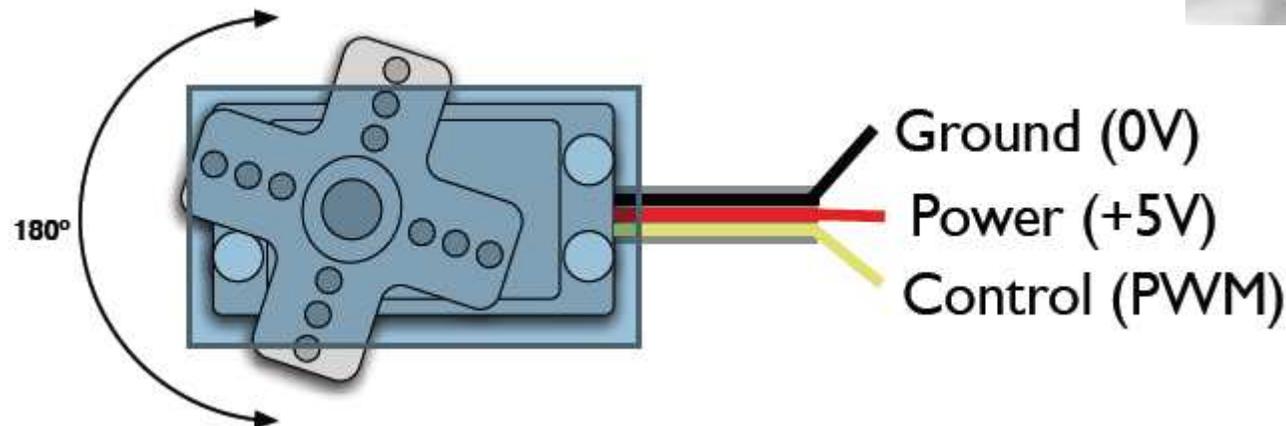


I servomotori dove si utilizzano?

- In robotica, negli effetti cinematografici, nei presepi e nei teatrini dei burattini si usano estensivamente.
- Ogni volta che avete bisogno di effettuare un piccolo controllo, un movimento ripetibile.
- La rotazione dell'alberino può essere trasformata in un movimento lineare con un semplice circuito meccanico.

I servomotori

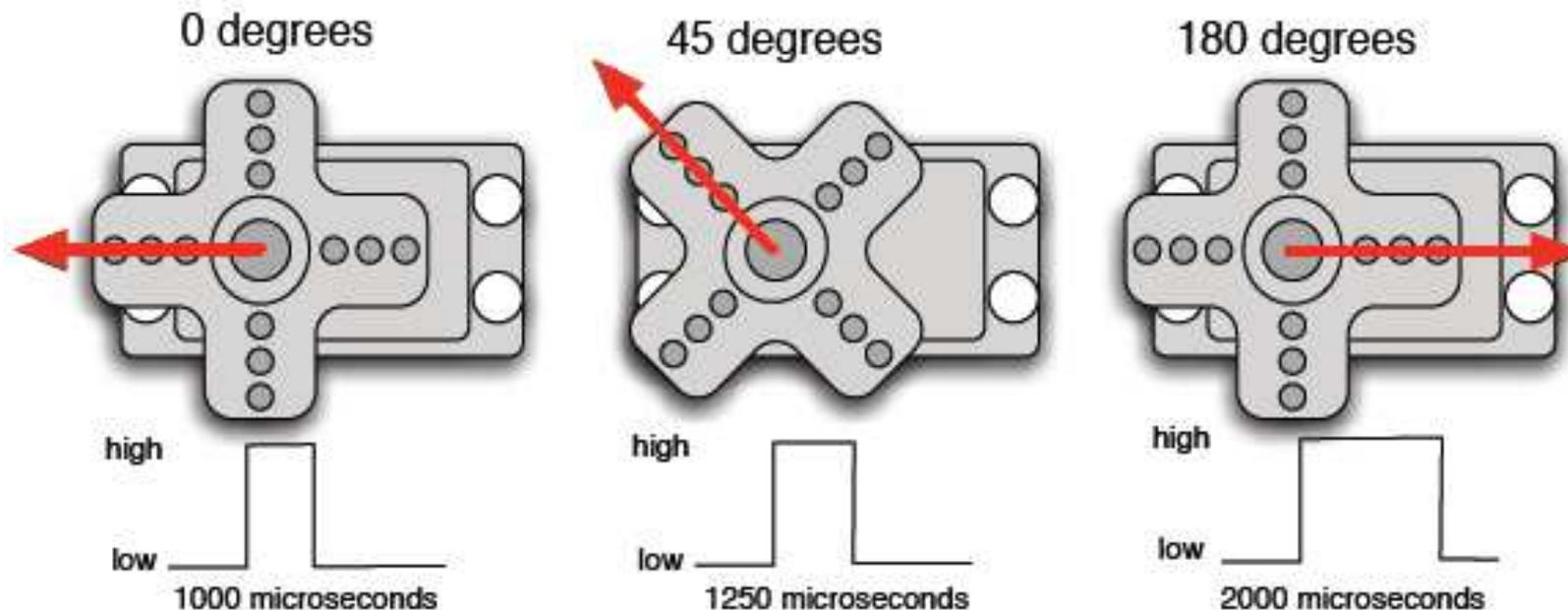
- Sono disponibili in molte dimensioni
- Dai modelli più piccoli (9 grammi)
- A quelli più grandi per le auto (157 grammi)
- TUTTI però possiedono solo 3 cavi di collegamento.
- La frequenza di aggiornamento del PWM è di 50 Hz (ogni 20 msec.)
- L'impulso varia da 1 a 2 msec.
- **1 msec** = posizione dell'alberino completamente ruotata con senso antiorario
- **2 msec** = posizione dell'alberino completamente ruotata con senso orario



Movimento dei servomotori

- Per posizionare il servomotore occorre trasmettere una serie di impulsi della durata da 1 a 2 msec.
- Per mantenere la posizione occorre ripetere periodicamente la trasmissione dell'impulso.
- E' necessario un certo tempo per ruotare. Se gli impulsi sono troppo veloci l'alberino non si sposta.

Ricorda → 1 msec. = 1000 μsec.

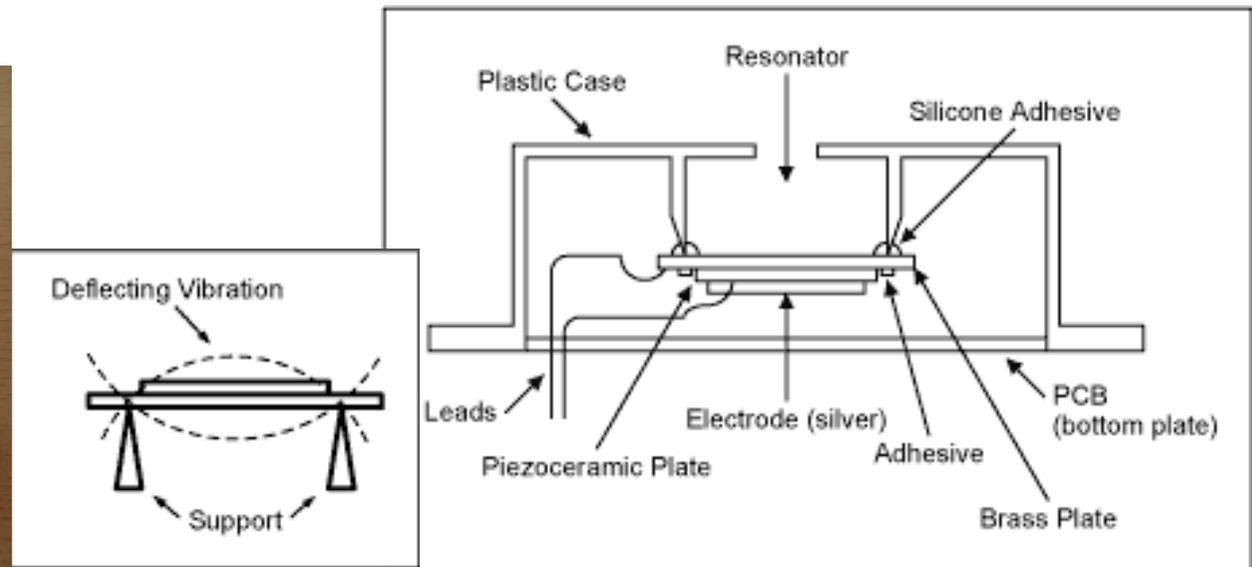


Buzzer piezoelettrico

- La parola “**piezein**” in greco significa "spremere"
Alcuni cristalli, quando sono sottoposti ad una forte pressione, creano una scintilla.
- Si è scoperto che il processo è reversibile, cioè va anche nel senso opposto quando viene sottoposto ad una tensione ai suoi capi si flette per emettere il suono (flettere qualcosa avanti e indietro permette di muovere l'aria circostante emettendo un suono)

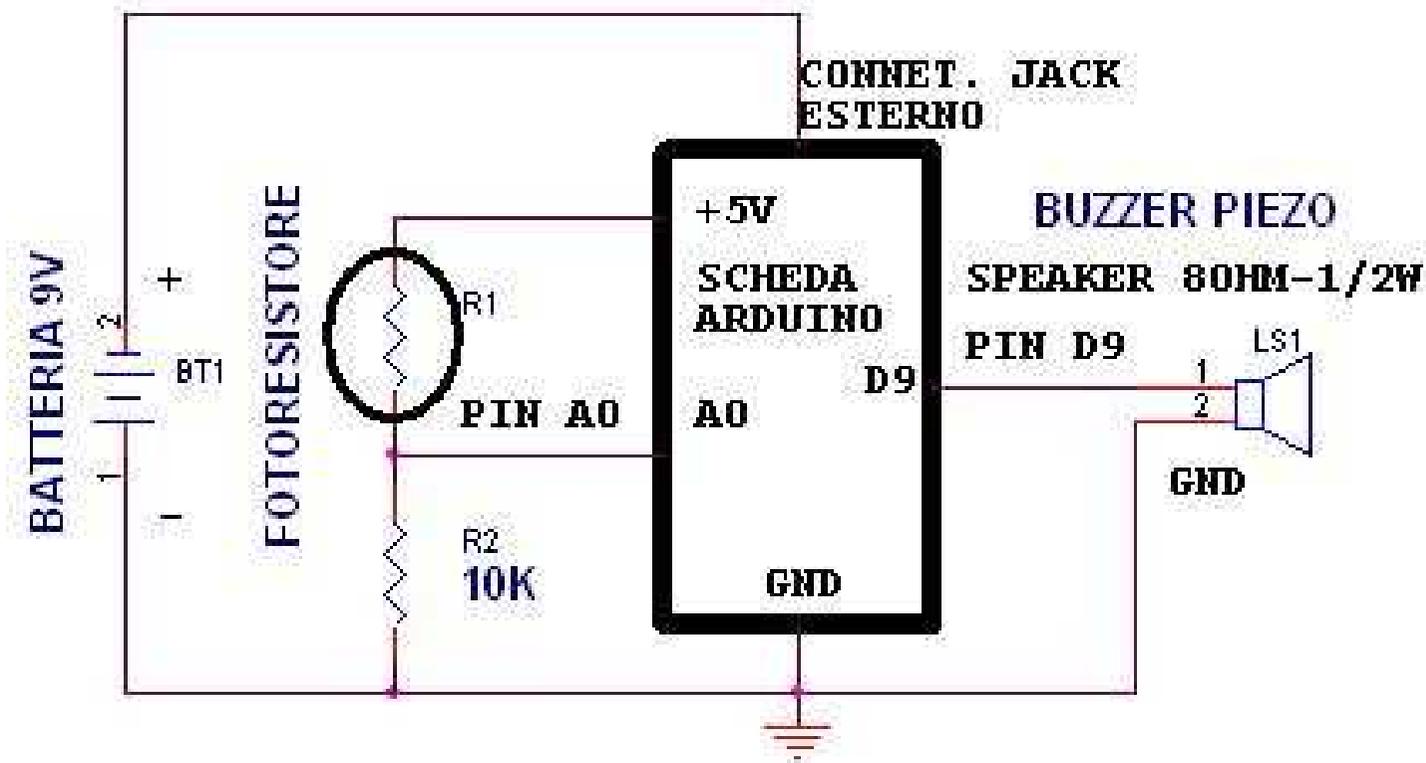
Buzzer piezoelettrici e altoparlanti

- Collegamento tramite 2 conduttori.
- Basta applicare una tensione oscillante per ottenere un suono
- Il buzzer supporta l'elemento piezoelettrico ed ha una cavità risonante per il suono.
- Occorre applicare una tensione fluttuante perché se vien utilizzata solo un livello costante “high” oppure “low” non funzionerà.



Allarme DIN-DON per apertura cassetto o frigorifero

Si richiede l'implementazione di un software in grado di:
Generare due note (din-don) tramite buzzer o altoparlante
quando viene illuminata dalla luce ambiente la
fotoresistenza (uscita digitale ad onda quadra)



Allarme DIN-DON per apertura cassetto o frigorifero

/* Allarme DIN-DON per apertura cassetto o frigorifero
collegare un altoparlante da 8 ohm da un terminale sul pin digitale D9
e l'altro terminale a GND. Il fotoresistore R1 tra +5V e l'ingresso
analogico A0, mentre R2 = 10Kohm sara' collegato tra A0 e GND.
La batteria da 9V viene collegata al jack con il polo positivo centrale */

```
void setup()  
{  
    pinMode(9, OUTPUT);  
}  
  
void loop()  
{  
    // leggi il valore della tensione sull'input analogico A0  
    // finche' tale valore sara' inferiore a circa il 20% di 5V  
    // continua a rileggere l'input, altrimenti prosegui  
    // perche' la fotoresistenza e' stata colpita dalla luce  
    while(analogRead(0) < 200);  
    // genera una frequenza di 2000Hz sul pin 9 per 1000ms  
    tone(9, 2000, 1000);  
    delay(1000); // aspetta 1000ms  
    // genera una frequenza di 440Hz sul pin 9 per 2000ms  
    tone(9, 440, 2000);  
    delay(2000); // aspetta 2000ms  
    // blocca il generatore di frequenza sul pin 9:  
    noTone(9); // disabilita l'altoparlante  
}
```

allarme_DIN_DON.ino

Misura della temperatura – LM35

Low Cost

Sensore tipo **LM35** Precision Centigrade Temperature Sensors

Range di temperatura: da -55 a $+150^{\circ}\text{C}$ (contenitore plastico tipo TO-92)

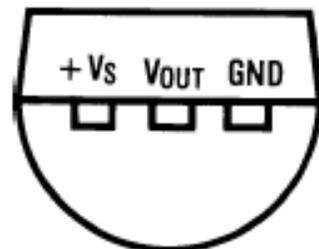
Accuratezza: $\pm 3/4^{\circ}\text{C}$ su tutta la scala, $\pm 1/2^{\circ}\text{C}$ ambiente)

Bassa impedenza di uscita (0.1Ω per 1 mA carico)

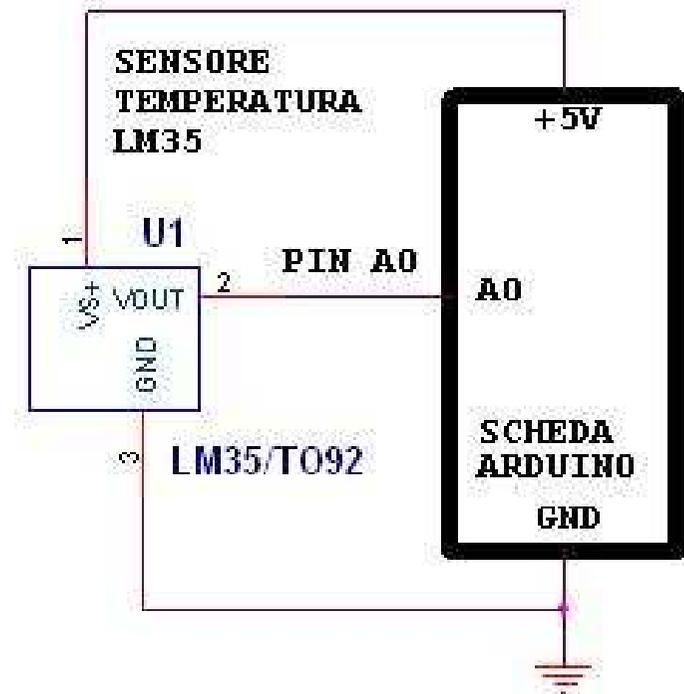
Uscita in tensione lineare $+10.0 \text{ mV}/^{\circ}\text{C}$

Alimentazione tra $4 \div 30\text{V}$

TO-92
Plastic Package



BOTTOM VIEW



Visto di sotto dal lato terminali

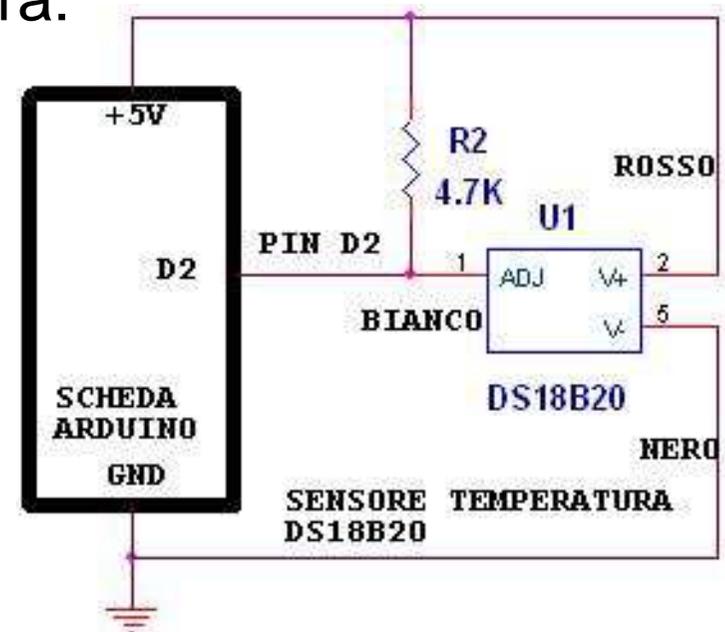
Misura della temperatura - LM35

```
int pin_sensore = 0; // pin 0 di lettura tensione analogica
int temperatura = 0; // variabile temperatura
void setup()
{
  Serial.begin(9600); // inizializza seriale a 9600 baud
}
void loop()
{ // Conversione della tensione generata dal sensore di temperatura
  temperatura = ( 5.0 * analogRead(pin_sensore) * 100.0) / 1024.0;
  Serial.print("Temperatura: "); // trasmetti la scritta "Temperatura: "
  Serial.println(temperatura); // trasmetti il dato della temperatura
  delay(1000); // ritardo tra una lettura e l'altra di 1 sec.
}
```

temperatura_LM35.ino

Misura della temperatura – DS18B20

- Sensore tipo **DS18B20 (1-Wire Digital Thermometer)**
- Range di temperatura: da -55 a $+125^{\circ}\text{C}$ (contenitore waterproof completamente isolato)
- Risoluzione: programmabile da 9 a 12 bit, accuratezza $\pm 1/2^{\circ}\text{C}$ ambiente
- Uscita digitale: livello TTL compatibile con lettura fino a 32 sensori differenti collegati sulla stessa linea
- Alimentazione tra $3,3 \div 5\text{V}$
- Tempo di conversione della temperatura:
750 msec. max. a 12 bit
valori compresi tra $0 \div 4095$.



Misura della temperatura – DS18B20

```
#include <OneWire.h>
int DS18S20_Pin = 2; //DS18S20 Signal pin on digital 2
//Temperature chip i/o
OneWire ds(DS18S20_Pin); // on digital pin 2
void setup(void) { Serial.begin(9600); }
void loop(void)
{
  float temperature = getTemp();
  Serial.println(temperature);
  delay(100); //just here to slow down the output so it is easier to read
}
float getTemp()
{ //returns the temperature from one DS18S20 in DEG Celsius
  byte data[12];
  byte addr[8];
  if ( !ds.search(addr) )
  { //no more sensors on chain, reset search
    ds.reset_search();
    return -1000;
  }
  if ( OneWire::crc8( addr, 7) != addr[7] )
  {
    Serial.println("CRC is not valid!");
    return -1000;
  }
}
```

Sensore_Temperatura_DS18B20.ino

Parte 1^a

Misura della temperatura – DS18B20

```
if ( addr[0] != 0x10 && addr[0] != 0x28)
{
    Serial.print("Device is not recognized");
    return -1000;
}
ds.reset();
ds.select(addr);
ds.write(0x44,1); // start conversion, with parasite power on at the end
byte present = ds.reset();
ds.select(addr);
ds.write(0xBE); // Read Scratchpad
for (int i = 0; i < 9; i++)
{ // we need 9 bytes
    data[i] = ds.read();
}
ds.reset_search();
byte MSB = data[1];
byte LSB = data[0];
float tempRead = ((MSB << 8) | LSB); //using two's compliment
float TemperatureSum = tempRead / 16;
return TemperatureSum;
}
```

Sensore_Temperatura_DS18B20.ino

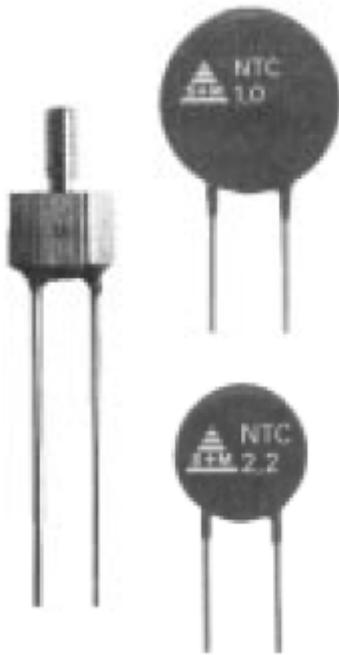
Parte 2^a

OUTPUT

26.62
26.62
26.62
26.81
26.81
26.81
26.81
26.81
26.81
27.00
27.00
27.00
27.00
27.00
27.00
27.19
27.19
27.19
27.19
27.19
27.37
27.37
27.37
27.37
27.37
27.37
27.56
27.56

Misura della temperatura con Termistori NTC

- Come definizione un termistore di tipo **NTC** (**Negative Temperature Coefficient**) è un resistore metallico sensibile alla temperatura che fornisce una diminuzione del valore resistivo nominale in conseguenza di un incremento della temperatura.
- Con un valore compreso tra $-2\%/K$ a $-6\%/K$, si ha un coefficiente negativo della temperatura di circa 10 volte maggiore degli altri metalli.
- L'elevata sensibilità dei termistori NTC è ideale nelle applicazioni di misura della temperatura in cui si richiede un basso costo.
- I sensori NTC vengono utilizzati generalmente entro un range di misura della temperatura compreso tra -40 e $+300\text{ }^{\circ}\text{C}$.



Misura della temperatura con Termistori NTC

```
/* I.I.S. Primo LEVI - Torino
   Progetto: NTC.ino      Autore: Questo è un esempio di pubblico
   Descrizione: Lettura di un input analogico collegato al sensore NTC con il pin A0 e
   l'altro a +5V, mentre un resistore da 1 Kohm verra' collegato tra A0 e massa.
   La visualizzazione della temperatura e' in gradi centigradi sulla seriale */
#include <math.h>
#define ThermistorPIN 0 // collegato al sensore NTC da 5 Kohm Pin analogico 0
float val = 0; // Variabile per il calcolo della temperatura in base al proprio NTC
float NTC = 5000; // Valore in ohm dell'NTC utilizzato. 5000=5k
float Thermistor(int RawADC)
{
    val = 10000 / NTC;
    val = 4.5 * val;
    val = NTC * val;
    long Resistance;
    float Temp; // variabile per il calcolo della temperatura

    Resistance=((1024 * val / RawADC) - val);
    Temp = log(Resistance); // salva Long(resistance)
    Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp * Temp));
    Temp = Temp - 273.15; // converte gradi Kelvin in Celsius
    return Temp; // restituisci la temperatura calcolata
}
```

NTC.ino - 1^a parte

Misura della temperatura con Termistori NTC

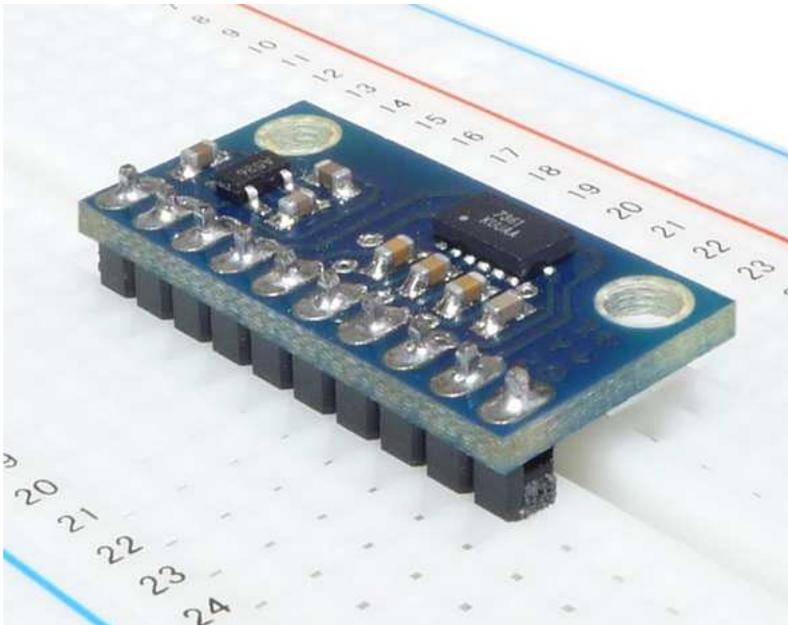
NTC.ino - 2^a parte

```
void setup()
{
  Serial.begin(9600); // settaggio velocita' seriale
}

void loop()
{
  float temp1;
  temp1=Thermistor(analogRead(ThermistorPIN)); // legge valore ADC e converte in °C
  Serial.print("Temperatura: ");
  Serial.print(temp1,1); // Visualizzazione in gradi Celsius
  Serial.write(176);
  Serial.print("C");
  Serial.print(" / ");
  temp1 = (temp1 * 9.0) / 5.0 + 32.0; //Converte in Fahrenheit
  Serial.print(temp1,1); // Visualizza in gradi Fahrenheit
  Serial.write(176);
  Serial.println("F");
  delay(1000); // ritardo di 1sec.
}
```

Misura della accelerazione con sensore MMA7361L su 3 assi X, Y e Z

Sensibilità regolabile a $\pm 1.5g$ (porre l'input g-Select = LOW = 0V) oppure a $\pm 6g$ (g-Select = HIGH = 3,3V)



Vdd = +3,3V

Vss = GND

Xout = uscita analogica in mV relativa all'accelerazione dell'asse **X**

Yout = uscita analogica in mV relativa all'accelerazione dell'asse **Y**

Zout = uscita analogica in mV relativa all'accelerazione dell'asse **Z**

Sensore MMA7361L collegamenti

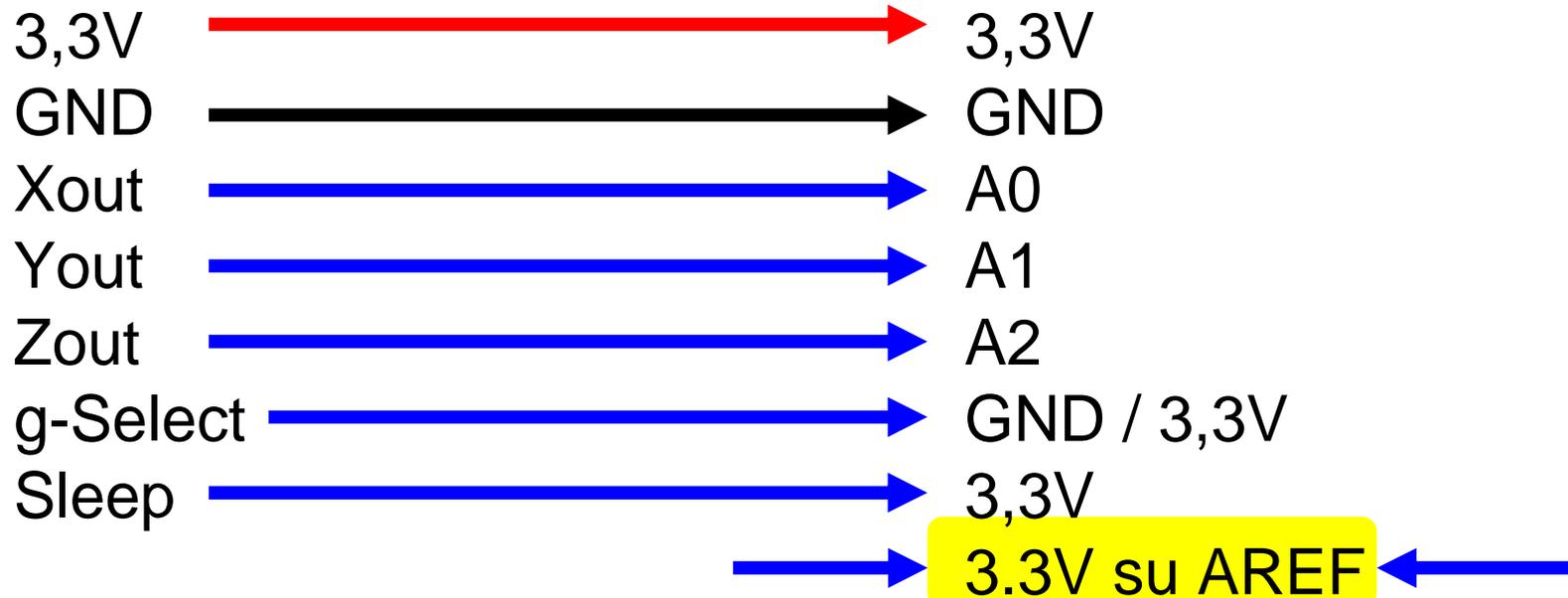
Con la sensibilità impostata a $\pm 1.5g$ (porre l'input g-Select = **LOW** = **0V**) l'accelerazione si può misurarla come variazione dei mV sui singoli pin di out il cui valore varia di **800 mV** ad ogni variazione di **1g** (g = accelerazione di gravità).

Se la sensibilità è impostata a $\pm 6g$ (g-Select = **HIGH** = **3,3V**) la variazione **ad ogni g** è di soli **206 mV**.

COLLEGAMENTI

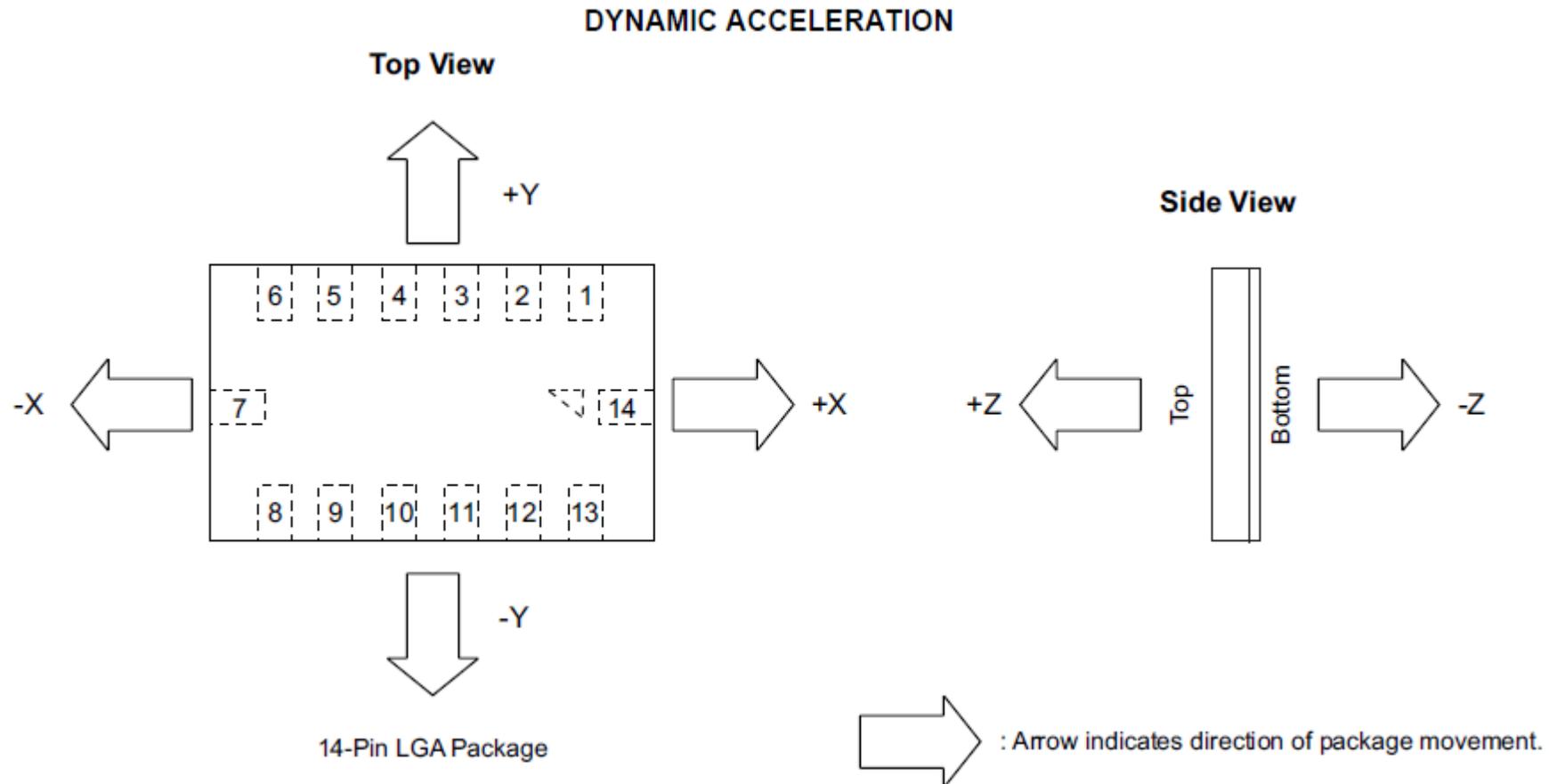
accelerometro MMA7361L

Arduino



Misura della accelerazione con sensore MMA7361L su 3 assi X, Y e Z

Il datasheet dell'accelerometro MMA7361L indica la direzione di ogni asse rispetto all'integrato per aiutarci a comprendere ad ogni movimento gli assi corrispondenti che vengono modificati.



Sensore MMA7361L software

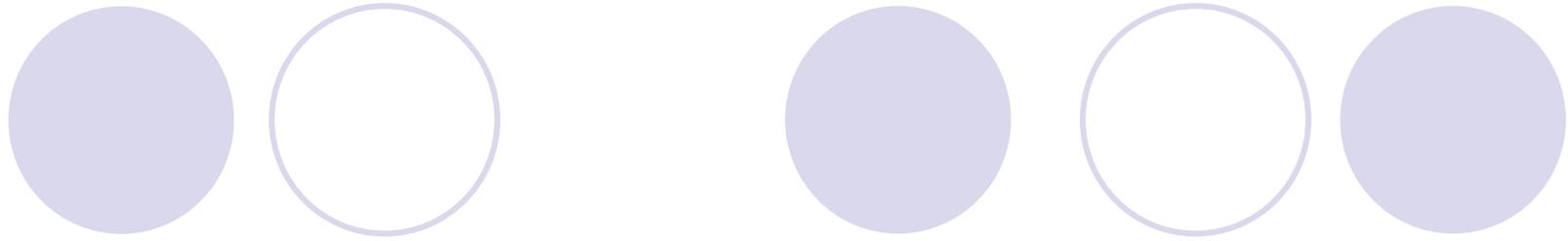
```
/* I.I.S. Primo LEVI - Torino    Data: 03/03/2013
   Progetto: accelerometro_3_assi_MMA7361L    Autore: G. Carpignano
   Descrizione: Scrivere un programma in modo tale che venga letto il sensore MMA7361L
   in grado di restituire l'accelerazione sui 3 assi */

int X_pin = A0; // pin analogico dell'Asse X
int Y_pin = A1; // pin analogico dell'Asse Y
int Z_pin = A2; // pin analogico dell'Asse Z

void setup() // inizializzazione, viene eseguito una sola volta all'inizio
{ // inizializza la seriale RS232 con 9600 baud, 8 bit dati, nessuna parità e 1 bit di stop
  Serial.begin(9600);
  analogReference(EXTERNAL); // pin AREF (riferimento analogico esterno) da collegare a 3,3V
  Serial.println("\nInizializzazione del sensore MMA7361L");
}

void loop() // programma principale
{
  Serial.print("X: ");
  Serial.print(analogRead(X_pin) ); // leggi e stampa l'input analogico A0 dell'asse X
  Serial.print("  Y: ");
  Serial.print(analogRead(Y_pin) ); // leggi e stampa l'input analogico A1 dell'asse Y
  Serial.print("  Z: ");
  Serial.print(analogRead(Z_pin) ); // leggi e stampa l'input analogico A2 dell'asse Z
  Serial.print("\n"); // stampa un CR (Carriage Return --> ritorno a inizio riga)
  delay(100); // ritardo espresso in msec.
}
```

Accelerometro_3_assi_MMA7361L.ino



Fine lavori

