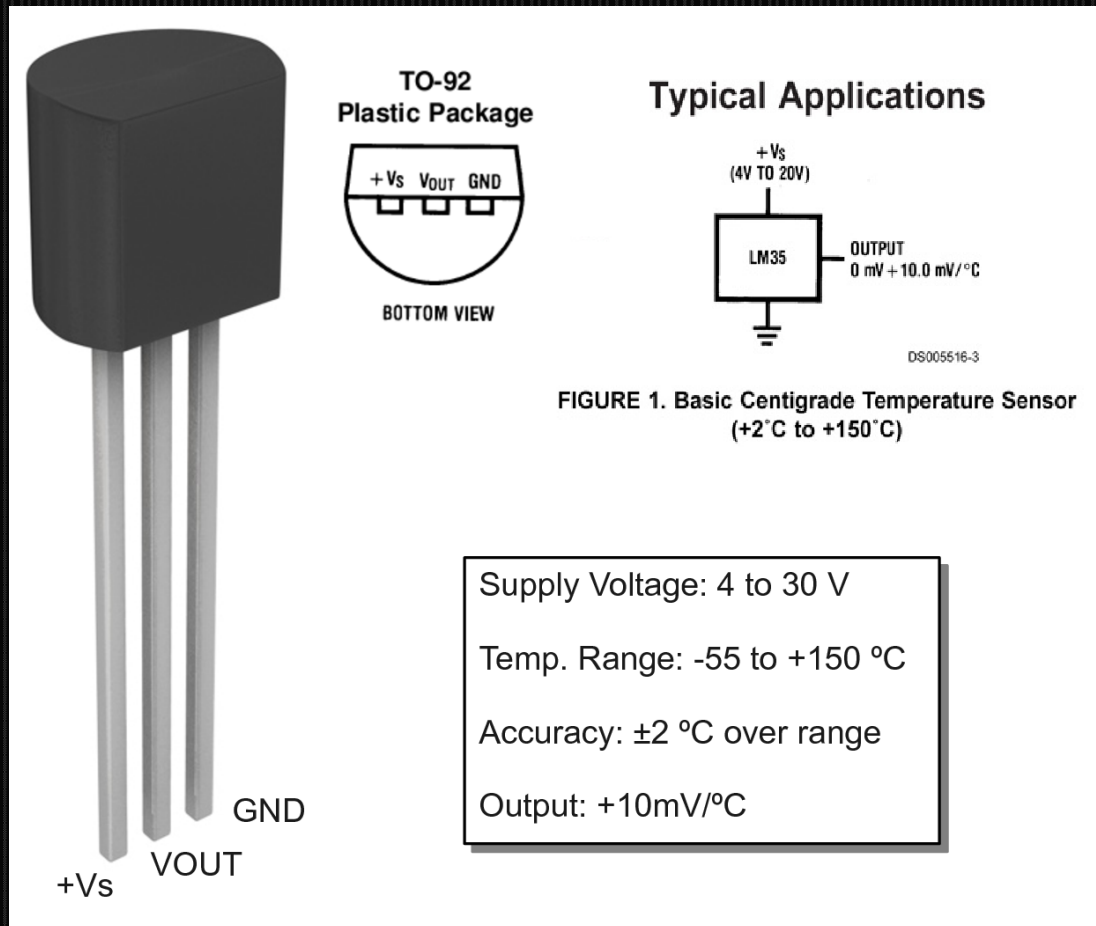


Progetti con arduino

LM35

Caratteristiche del trasduttore



Caratteristiche del trasduttore

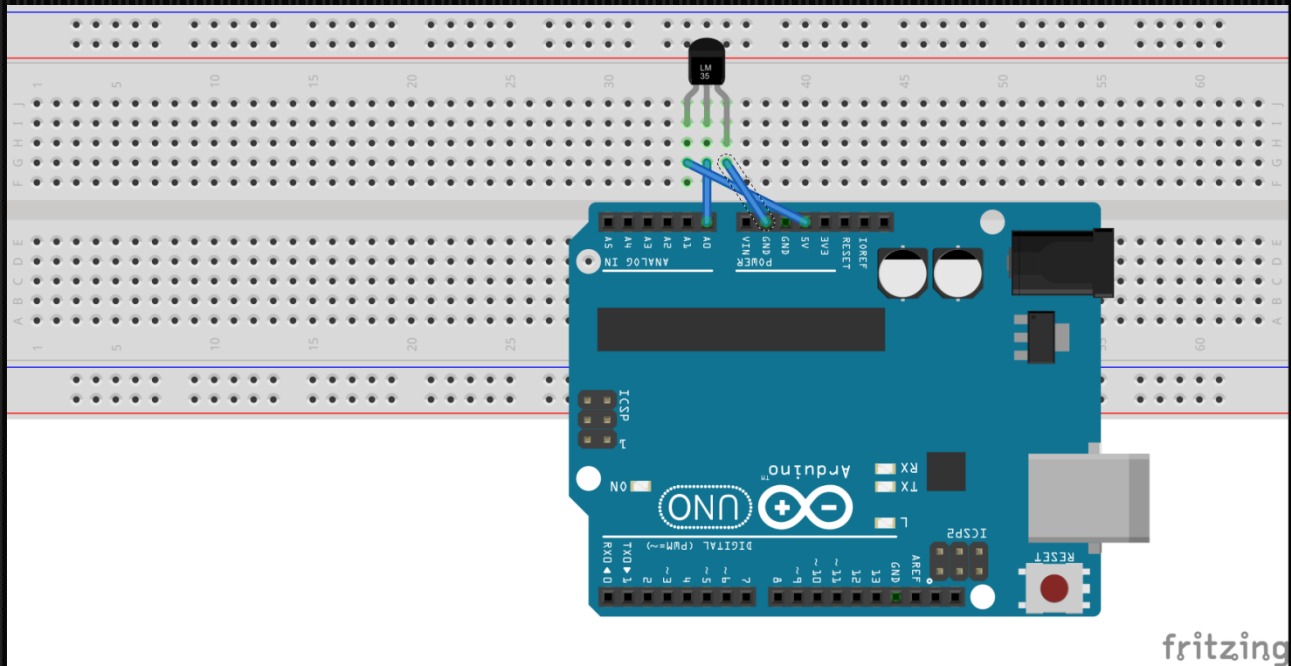
- Il trasduttore è analogico
- Trasforma la temperatura in tensione secondo la relazione

$$V_{\text{millivolts}} = 10 * t_{\text{celsius}}$$
$$100 * V_{\text{volts}} = t_{\text{celsius}}$$

Essendo V la d.d.p. e t la temperatura.

- $-50^{\circ}\text{C} < t < 150^{\circ}\text{C}$
- La precisione è $\pm 0.5^{\circ}\text{C}$
- La tensione di alimentazione è tra 4 V e 30 v

Circuito



Corpo del programma

```
float sum=0.0;
float media=0.0;
int i=0;
for(i=0;i<5;i++)
{
  sensor = analogRead(A0);
  v = ( sensor/1024.0)*5000;
  temp=v/10;
  sum=sum+ temp;)
  delay(200); //ritardo tra due campionature successive
}
media= (sum/5); //calcolo del valore medio di 5 rilievi
Serial.print(media);Serial.println(" C"); //stampa su serial
//monitor del valore di temperatura in gradi Celsius
```

Lm35 sensore analogico

- Lm35 è analogico e lineare
- Affinchè i dati acquisiti dal sensore Lm35 possano essere analizzati dal microcontrollore, è necessario fare una conversione A/D.
- Il convertitore A/D di Arduino è a 10 bit, 1023 livelli
- Nella conversione A/D a 10 bit, il valore digitale elaborato dal microcontrollore vale:
 $V = V_{ref} * V_{in} / 1023$
- $T = 100 * V_{ref} * V_{in} / 1023$

Programma per salvare su una eeprom

```
#define LM35_pin 1
#include <EEPROM.h>

// the current address in the EEPROM (i.e. which byte
// we're going to write to next)
int addr = 0;
/* Definizioni globali */
float vref = 1.1; // Vref dell'ADC (quell'interno è di 1,1V)
/* Impostazione dell'hardware */
void setup()
{
  analogReference( INTERNAL ); // per l'ADC usiamo il Vref interno da 1,1V (migliore precisione)
  analogRead( LM35_pin ); // Prima lettura "a vuoto" (serve per l'assestamento dell'ADC)
  Serial.begin(9600); // Impostazione per l' LCD (2x16)
}
void loop()
{
  // invia il valore al LCD
  float temp = 0.0; // valore convertito in temperatura (°C)
  int val = 0; // valore quantizzato dall'ADC [0..1023]
  int nread = 5; // numero di letture (da 5 a 8)
  float somma = 0.0;
  float temperatura; // somma delle letture
```

Programma per salvare su una eeprom

```
for (int i=0; i<nread; i++)
{
    val = analogRead( LM35_pin );           // legge il dato della tensione sul pin
    'LM35_pin'
    temp = ( 100.0 * vref * val ) / 1024.0; // lo converte in °C
    somma += temp;                          // aggiunge alla somma delle temperature lette
}
temperatura=somma/nread;
EEPROM.write(addr, temperatura);
Serial.println(temperatura);
// advance to the next address. there are 512 bytes in
// the EEPROM, so go back to 0 when we hit 512.
addr = addr + 1;
if (addr == 512)
    addr = 0;
delay(300);
}
```


Programma di processing

```
import processing.serial.*;

Serial myPort;    //The serial port
int xPos = 1;     // horizontal position of the graph

void setup () {
  // set the window size:
  size(400, 300);

  // List all the available serial ports
  println(Serial.list());
  // I know that the first port in the serial list on my mac
  // is always my Arduino, so I open Serial.list()[0].
  // Open whatever port is the one you're using.
  myPort = new Serial(this, Serial.list()[0], 9600);
  // don't generate a serialEvent() unless you get a newline character:
  myPort.bufferUntil('\n');
  // set initial background:
  background(0);
}

void draw () {
  // everything happens in the serialEvent()
}
```

Programma di processing

```
void serialEvent (Serial myPort) {  
  // get the ASCII string:  
  String inString = myPort.readStringUntil('\n');  
  
  if (inString != null) {  
    // trim off any whitespace:  
    inString = trim(inString);  
    // convert to an int and map to the screen height:  
    float inByte = float(inString);  
    inByte = map(inByte, 0, 1023, 0, height);  
  
    // draw the line:  
    stroke(127,34,255);  
    line(xPos, height, xPos, height - inByte*20);  
  
    // at the edge of the screen, go back to the beginning:  
    if (xPos >= width) {  
      xPos = 0;  
      background(0);  
    }  
    else {  
      // increment the horizontal position:  
      xPos++;  
    }  
  }  
}
```

Controllo della temperatura con una ventola

```
#define LM35_pin 1
/* Definizioni globali */
float vref = 1.1; // Vref dell'ADC (quell'interno è di 1,1V)
/* Impostazione dell'hardware */
void setup()
{
  analogReference( INTERNAL ); // per l'ADC usiamo il Vref interno da 1,1V (migliore precisione)
  analogRead( LM35_pin ); // Prima lettura "a vuoto" (serve per l'assestamento dell'ADC)
  Serial.begin(9600);
  pinMode(3, OUTPUT);
}
void loop()
{
  // invia il valore al LCD
  float temp = 0.0; // valore convertito in temperatura (°C)
  int val = 0; // valore quantizzato dall'ADC [0..1023]
  int nread = 5; // numero di letture (da 5 a 8)
  float somma = 0.0;
  float temperatura; // somma delle letture
  for (int i=0; i<nread; i++)
  {
    val = analogRead( LM35_pin ); // legge il dato della tensione sul pin 'LM35_pin'
    temp = ( 100.0 * vref * val ) / 1024.0; // lo converte in °C
    somma += temp; // aggiunge alla somma delle temperature lette
  }
  temperatura=somma/nread;
  if(temperatura>25)analogWrite(3,255);
  Serial.println(temperatura);
  delay(6000);
}
```

Progetto controllo di temperatura

