

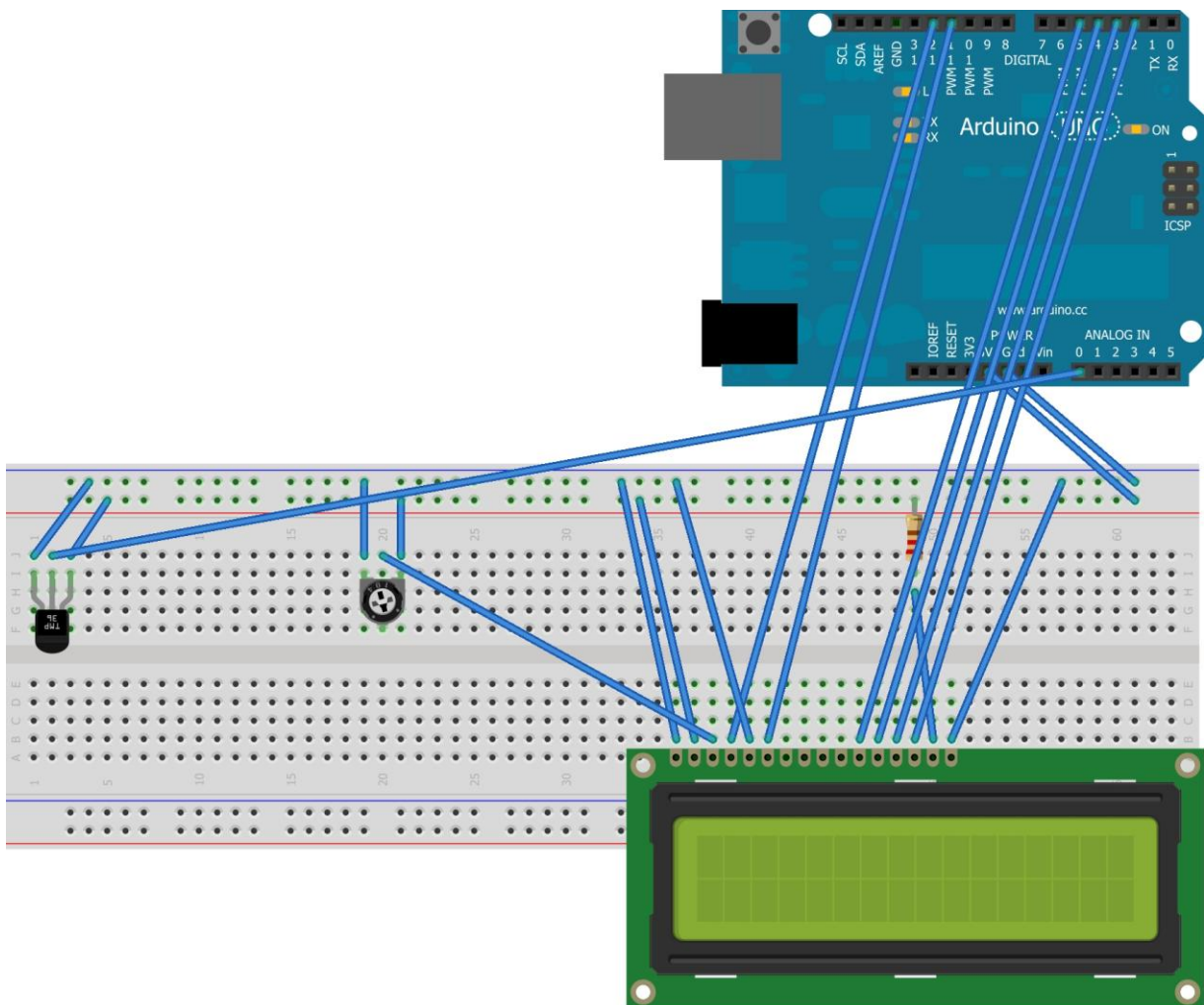
# Termometro con Arduino Uno

Ho realizzato questo termometro per misurare la temperatura in gradi Celsius e leggerla su un LCD assieme ad altri dati. La temperatura ha un range di errore massimo di  $\pm 2^{\circ}\text{C}$ , determinato dal sensore di temperatura (chip) del termometro.

## 1. Occorrente

- 1 Arduino Uno rev3 (vanno bene anche un Arduino Uno rev2, Duemilanove, Mega o con caratteristiche simili);
- 1 o più Breadboard Sperimentale/i;
- 1 Sensore di temperatura TMP36;
- 1 Potenziometro 10K $\Omega$ ;
- 1 resistenza da 220 $\Omega$ ;
- 1 LCD 16x2;
- 16 cavi jumper;
- 1 cavo USB;
- Computer con 1 porta USB, Arduino IDE installato (eseguibile su Windows, Mac OS X e Linux).

I collegamenti vanno fatti come nella foto/schema di seguito:



## 2. Codice

Dopo averlo assemblato ho scritto il codice di programmazione e lo ho caricato sull'arduino. Il codice da utilizzare è il seguente

```
#include <LiquidCrystal.h>

int Analog = 0;

float volt = 0;

float a = 0;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

  lcd.begin(16, 2);

}

void loop(){

  lcd.clear();

  Analog = analogRead(0);

  volt = ((Analog*4.610)/1024.00);

  a = ((volt-0.500)*100)+8.00;

  lcd.setCursor(0,0);

  lcd.print("Ana mV  *C");

  lcd.setCursor(0, 1);

  lcd.print(Analog);

  lcd.setCursor(4,1);

  lcd.print(volt);

  lcd.setCursor(9,1);

  lcd.print(a);

  delay (2000);

}
```

### 3. Spiegazione del codice

`#include <LiquidCrystal.h>` Questa espressione indica che, quando il codice verrà caricato, dovrà includere la libreria LiquidCrystall, ovvero la libreria per la gestione degli LCD.

`int Analog = 0;` Significa che l'arduino utilizzerà come pin di lettura del valore analogico il pin A0. Int segna una variabile di numero massimo +/-32.768. 0 si usa per lasciare la variabile così com'è.

`float volt = 0;` Inizializza una variabile come int, ma float si usa per il calcolo matematico, in quanto supporta numeri fino a +/-3,4028235<sup>38</sup>.

`float a = 0;` Vedi sopra.

`LiquidCrystal lcd(12, 11, 5, 4, 3, 2);`

Questo comando attiva l'LCD ed indica su quali pin è collegato.

`void setup() {` Il comando inizializza funzioni di impostazione generale. Le parentesi graffe delimitano queste impostazioni.

`lcd.begin(16, 2);` Vuol dire che l'LCD è un 16x2. Il comando lo comunica all'arduino.

`}` Fine void setup().

`void loop(){` Questa istruzione inizializza funzioni ripetitive fino all'infinito, può comprendere altre funzioni al suo interno e sono tutte delimitate da parentesi graffe

`lcd.clear();` Questa funzione, come tutte quelle col suffisso "lcd." ha effetto direttamente sull'LCD. In questo caso clear significa pulisci l'LCD.

`Analog = analogRead(0);` Dice semplicemente alla scheda di leggere il valore in entrata del pin A0 (analogico) e di memorizzarlo sulla memoria temporanea (RAM) come variabile. N.B. I dati sono espressi con valori compresi tra 0 e 1023.

`volt = ((Analog*4.610)/1024.00);`

Questa formula calcola i volt in uscita a partire dal valore rilevato sul pin analogico sopra citato e memorizza sulla RAM il dato come variabile. Gli Analog sono i valori analogici rilevati da 0 a 1023 che, se moltiplicati per il voltaggio effettivo utilizzato e divisi per 1024 (è la somma del numero dei valori totali), darà come risultato i millivolt corrispondenti al valore analogico.

`a = ((volt-0.500)*100)+8.00;`

Questa è la formula che trasforma i dati di uscita in millivolt, rilevati dallo strumento, in gradi Celsius; anche questo dato è memorizzato sulla RAM come variabile. L'espressione sottrae ai millivolt letti in uscita 0,5 volt e poi lo moltiplica per cento, a questo valore aggiungerà infine 8 gradi Celsius di calibrazione dello strumento. Questa formula è data dai valori che si ricavano durante l'utilizzo: 500mV equivalgono a 0°C, 510mV a 1°C, 600mV a 10°C e così 750mV corrispondono a 25°C. Ecco spiegata l'equazione.

<code>lcd.setCursor(0,0);</code>	Con questo comando l'arduino sposta il cursore per scrivere nella riga 1 colonna 1. N.B.: I valori nel codice sono tutti espressi da 0 a il numero massimo reale in questione-1 , per cui la colonna 16 viene scritta come 15 e la riga 1 come 0.
<code>lcd.print("Ana mV *C");</code>	Con questo comando l'arduino visualizza un/dei dato/i sull'LCD (i dati tra parentesi). In questo caso stampa la visualizza tra le virgolette.
<code>lcd.setCursor(0, 1);</code>	Con questo comando l'arduino sposta il cursore per scrivere nella riga 2 colonna 1.
<code>lcd.print(Analog);</code>	Visualizza il dato memorizzato nella variabile Analog.
<code>lcd.setCursor(4,1);</code>	Con questo comando l'arduino sposta il cursore per scrivere nella riga 2 colonna 5.
<code>lcd.print(volt);</code>	Visualizza il dato memorizzato nella variabile volt.
<code>lcd.setCursor(9,1);</code>	Con questo comando l'arduino sposta il cursore per scrivere nella riga 2 colonna 10.
<code>lcd.print(a);</code>	Visualizza il dato memorizzato nella variabile a (temperatura).
<code>delay (2000);</code>	Aspetta 2 secondi prima di continuare il codice.
<code>}</code>	Fine void loop(). Viene eseguito un reset delle variabili.