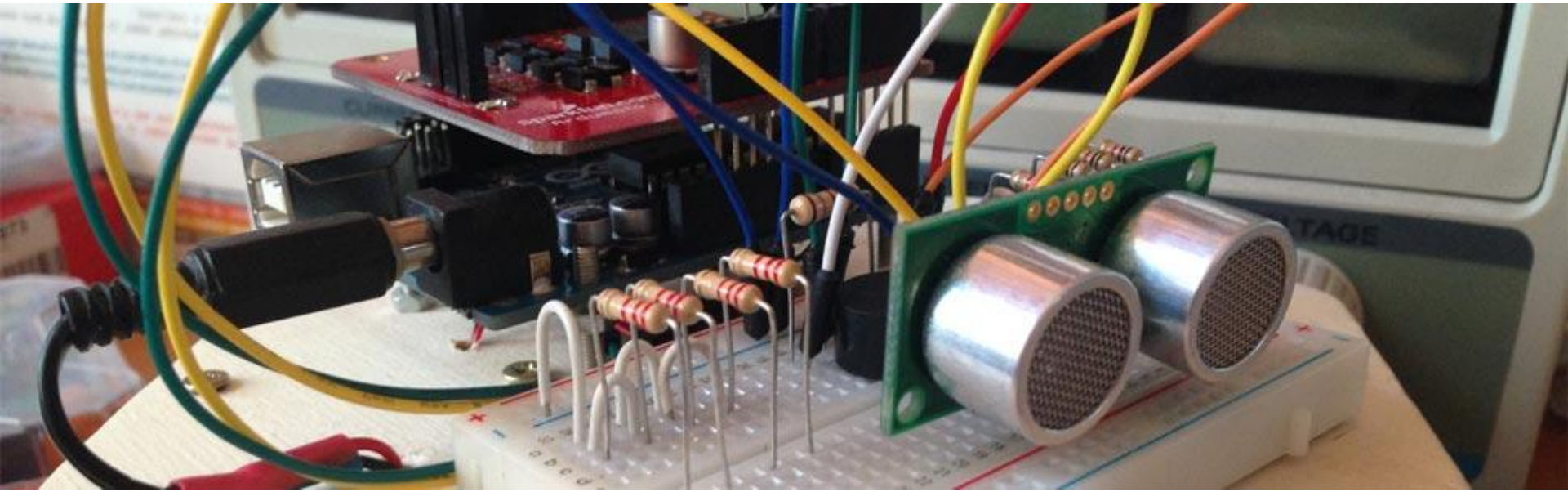
L'alfabeto di Arduino

Introduzione all'uso di Arduino
lezione 4

Prof. Michele Maffucci



Argomenti

- Input analogico - sensore di temperatura
- Uso del display LCD

Il codice e le slide utilizzate sono suscettibili di variazioni/correzioni che potranno essere fatte in ogni momento.

Introduzione

Prof. Michele Maffucci

Il seguente corso intende fornire le **competenze di base** per la realizzazione di lezioni di didattica delle robotica nella scuola secondaria di secondo grado.

Il corso ben si adatta a tutti i maker, studenti ed adulti, che per passione nell'elettronica necessitano di un'introduzione all'uso di Arduino.

Il docente che intendesse sviluppare un percorso didattico in cui si desidera realizzare dispositivi elettronici in grado di interfacciarsi col mondo fisico, potrà utilizzare queste lezioni come base per implementare moduli didattici aggiuntivi, pertanto questo corso è da intendersi come il mio personale tentativo di strutturare un percorso iniziale e modellabile a seconda del tipo di indirizzo della scuola. Chi vorrà potrà effettuare miglioramenti su quanto da me scritto.

Il percorso scelto è un estratto delle lezioni svolte durante i miei corsi di elettronica, sistemi ed impianti elettrici. Nelle slide vi sono cenni teorici di elettrotecnica che non sostituiscono in alcun modo il libro di testo, ma vogliono essere un primo passo per condurre il lettore ad un approfondimento su testi specializzati.

Il corso è basato sulla piattaforma Open Source e Open Hardware **Arduino** e fa uso dell'**Arduino starter kit**. Questa scelta non implica l'adozione di queste slide in corsi che non fanno uso di questo kit, ma è semplicemente una scelta organizzativa per lo svolgimento di questo corso di formazione. Alle proposte incluse nel kit ho aggiunto ulteriori sperimentazioni. Tutti i componenti possono essere acquistati separatamente.

Ulteriori approfondimenti e risorse a questo corso possono essere trovate sul mio sito personale al seguente link:

http://www.maffucci.it/area-studenti/arduino/

Nella sezione dedicata ad Arduino, sul mio sito personale, oltre ad ulteriori lezioni, di cui queste slide ne sono una sintesi, è possibile consultare un manuale di programmazione, in cui vengono dettagliate le istruzioni. Per rendere pratico l'utilizzo del manuale ne è stata realizzata anche una versione portable per dispositivi mobili **iOS** e **Android**, maggiori informazioni possono essere trovate seguendo il link.

Esempi utilizzati nel corso.

Tutti i programmi utilizzati nel corso possono essere prelevati al seguente link:

https://github.com/maffucci/LezioniArduino/tree/master/corso01

Gli sketch Arduino sono da scompattare nella cartella sketchbook.

Questo corso è nato in brevissimo tempo (circa 15 giorni) e quindi possibile che siano presenti delle imperfezioni, ringrazio fin d'ora chi vorrà segnalarmi correzioni e miglioramenti.

Per contatti ed ulteriori informazioni rimando alle ultime pagine di queste slide.

Grazie

Input analogico
sensore di temperatura

Prof. Michele Maffucci

rilevare temperature

Utilizziamo il sensore di temperatura per rilevare la temperatura del nostro corpo o dell'ambiente circostante. Utilizzeremo per questo esperimento il sensore TMP36, in grado di produrre in uscita una tensione direttamente proporzionale alla temperatura in gradi Celsius.

Dal datasheet si rileva che il sensore permette di misurare temperature comprese tra -40°C +125°C con precisione di 0,5°C, restituendo in uscita tensioni comprese tra 0,1 V e 1,7V (continui). Inoltre sempre dai datasheet si legge che una variazione di 1°C produce in uscita una tensione di 10 mV e alla temperatura di 0°C il sensore fornisce una tensione di 0,5 V

PIN 1, $+V_S$; PIN 2, V_{OUT}; PIN 3, GND

rilevare temperature

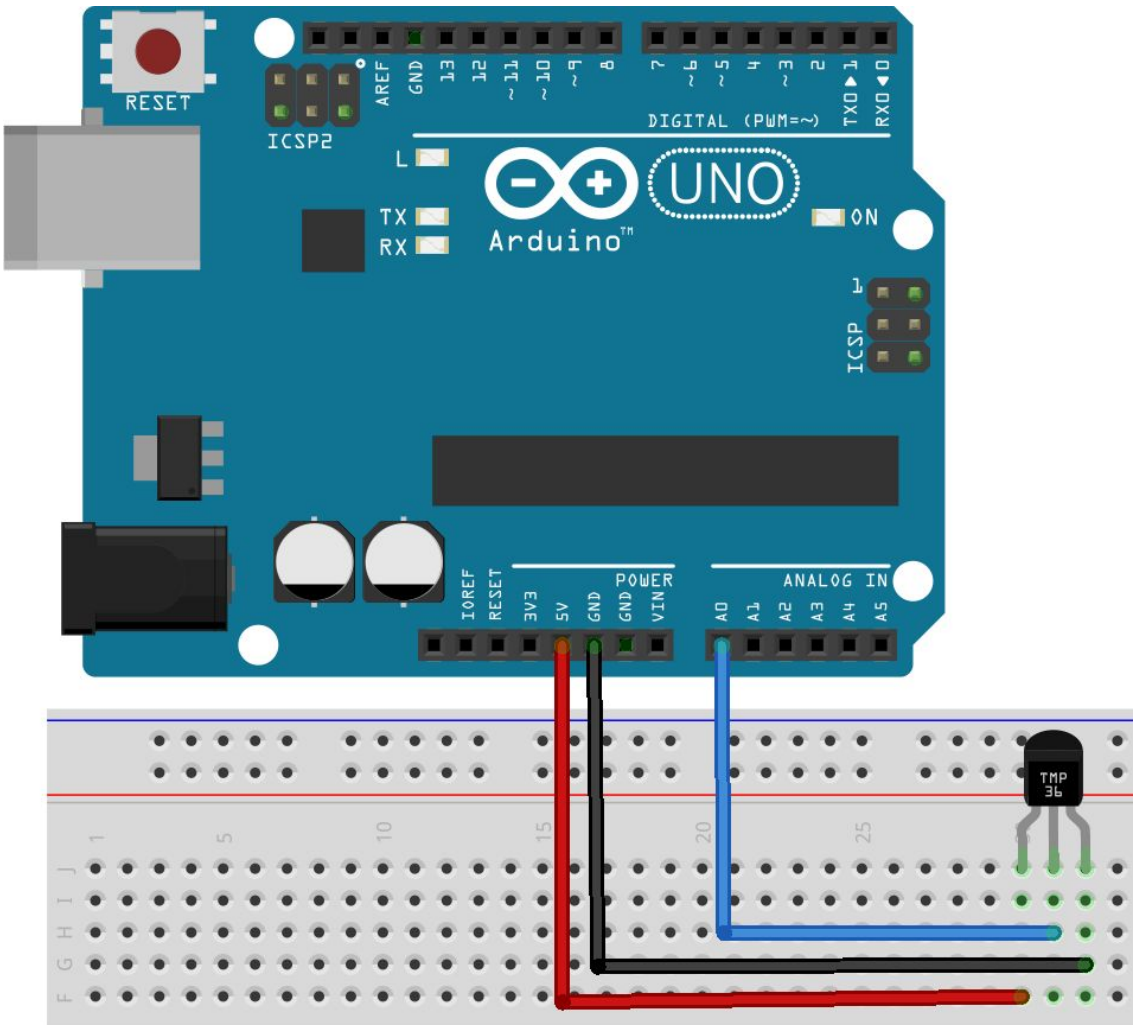
1/3

sketch31

Realizziamo un circuito per rilevare il valore letto dal sensore.
Visualizziamo questo valore sulla Serial monitor.

Componenti:

- Arduino
- TMP36



rilevare temperature | 2/5 | sketch31

```
/* Prof. Michele Maffucci
   25.03.2014

   Rilevare il valore letto dal
   sensore di temperatura
   TMP36

   Questo codice è di dominio pubblico

*/

void setup()
{
  Serial.begin(9600);          // inizializzazione della porta seriale

}

void loop()
{
  delay(1000);                 // si effettuano rilevazioni
                               // ogni secondo
  int valore = analogRead(0);
  Serial.println(valore);      // stampa sulla Serial monitor
                               // il valore letto dal sensore
}
```

Lo sketch converte in digitale il valore analogico letto sul pin 0 fornito dal TMP36

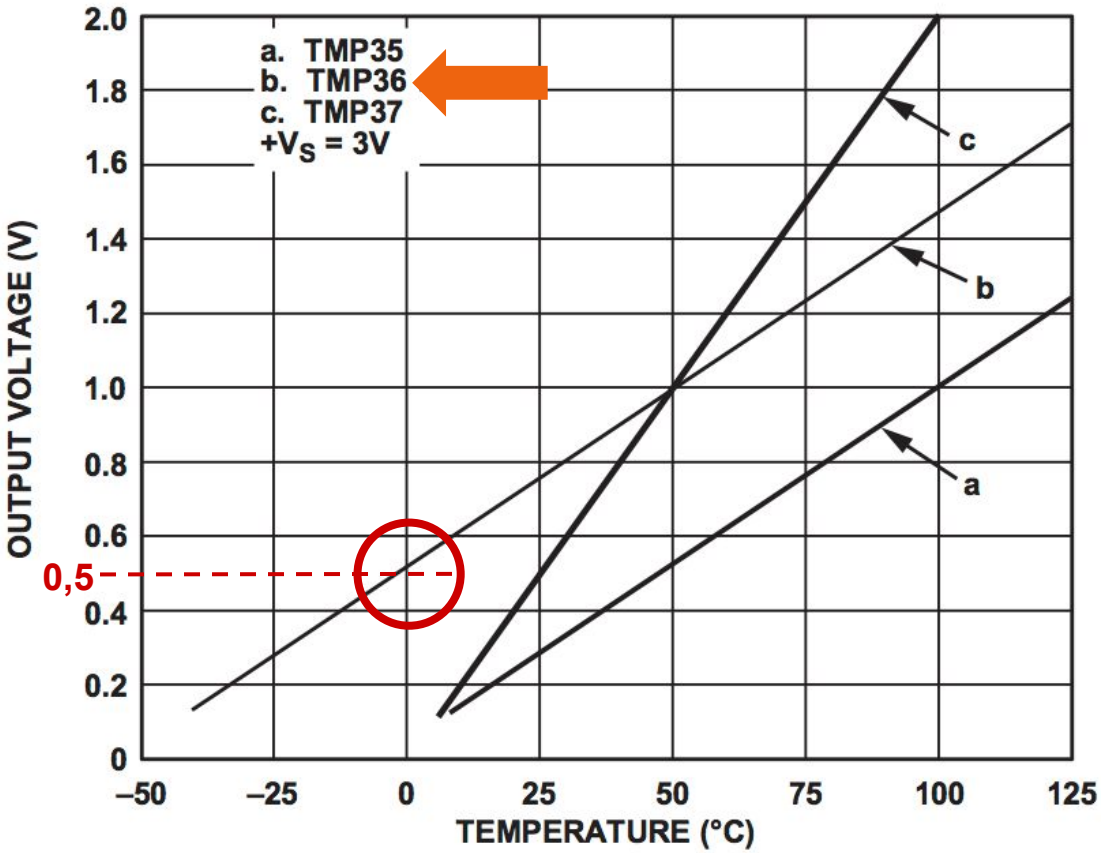
Il valore letto con il precedente skech deve ora essere convertito in una temperatura. Il datasheet del componente ci fornisce il grafico che mette in relazione la variazione di tensione con la temperatura °C/Vdc.

Dal grafico si osserva (retta b corrispondente al TMP36) che alla tensione di 0,5 V (500 mV) si ha una temperatura di 0°C, quindi tensioni inferiori a 0,5V indicano temperature negative.

Da quanto detto in precedenza sappiamo che **una variazione di 10 mV (0,01 V) implica una variazione di 1°C**, quindi se sul pin A0 si legge una tensione di 510 mV (0,51 V) vuol dire che il sensore sta rilevando una temperatura di:

510 mV - 500 mV = 10 mV

corrispondente a 1°C



Output Voltage vs. Temperature

Come precedentemente esposto l'**`analogRead(pin)`** legge il valore di tensione (compreso tra 0 e 5V) applicato sul piedino analogico 'pin' con una risoluzione di 10 bit e la converte in un valore numerico compreso tra 0 e 1023, corrispondente quindi ad un intervallo di 1024 valori.
Pertanto ogni intervallo corrisponde ad un valore di tensione V_u di:

$$V_u = \frac{5V}{1024} = 0{,}00488281\ V = 4{,}88\ mV\ \textit{(circa)}$$

Per sapere quindi il valore di tensione rilevato (nell'intervallo tra 0V e 5V) sarà sufficiente moltiplicare la tensione unitari V_u per il valore restituito dalla funzione **`analogRead(pin)`**, V_q (valore quantizzato) valore compreso tra 0 e 1023:

$$V_m = V_u \times V_q$$

Sapendo che V_u corrisponde a **4,88 mV**

possiamo anche scrivere che:

$$V_m = 4{,}88mV \times V_q$$

Da quanto detto per il sensore TMP36 sappiamo che una variazione di 10 mV corrisponde ad una variazione di 1°C, ciò vuol dire che se dividiamo la tensione misurata sul pin analogico, $\mathbf{V_m}$ per il valore di tensione corrispondente ad 1°C (10 mV=0,01V) e a questo valore sottraiamo il rapporto tra la tensione a 0°C (0,5V) per l'incremento unitario (10mV), si ottine la temperatura rilevata dal sensore:

$$\textbf{Temperatura} = \frac{\mathbf{V_m}}{0{,}01\ V} - \frac{0{,}5\ V}{0{,}01\ V} = \frac{\mathbf{V_m}}{0{,}01\ V} - \frac{5 \times 10^{-1} V}{1 \times 10^{-2}\ V} =$$

$$= \mathbf{V_m} \times \mathbf{100} - 50 = 4{,}88mV \times \mathbf{V_q} \times 100 - 50 = 0{,}488 \times \mathbf{V_q} - 50$$

$$\mathbf{V_m} = 4{,}88mV \times \mathbf{V_q}$$

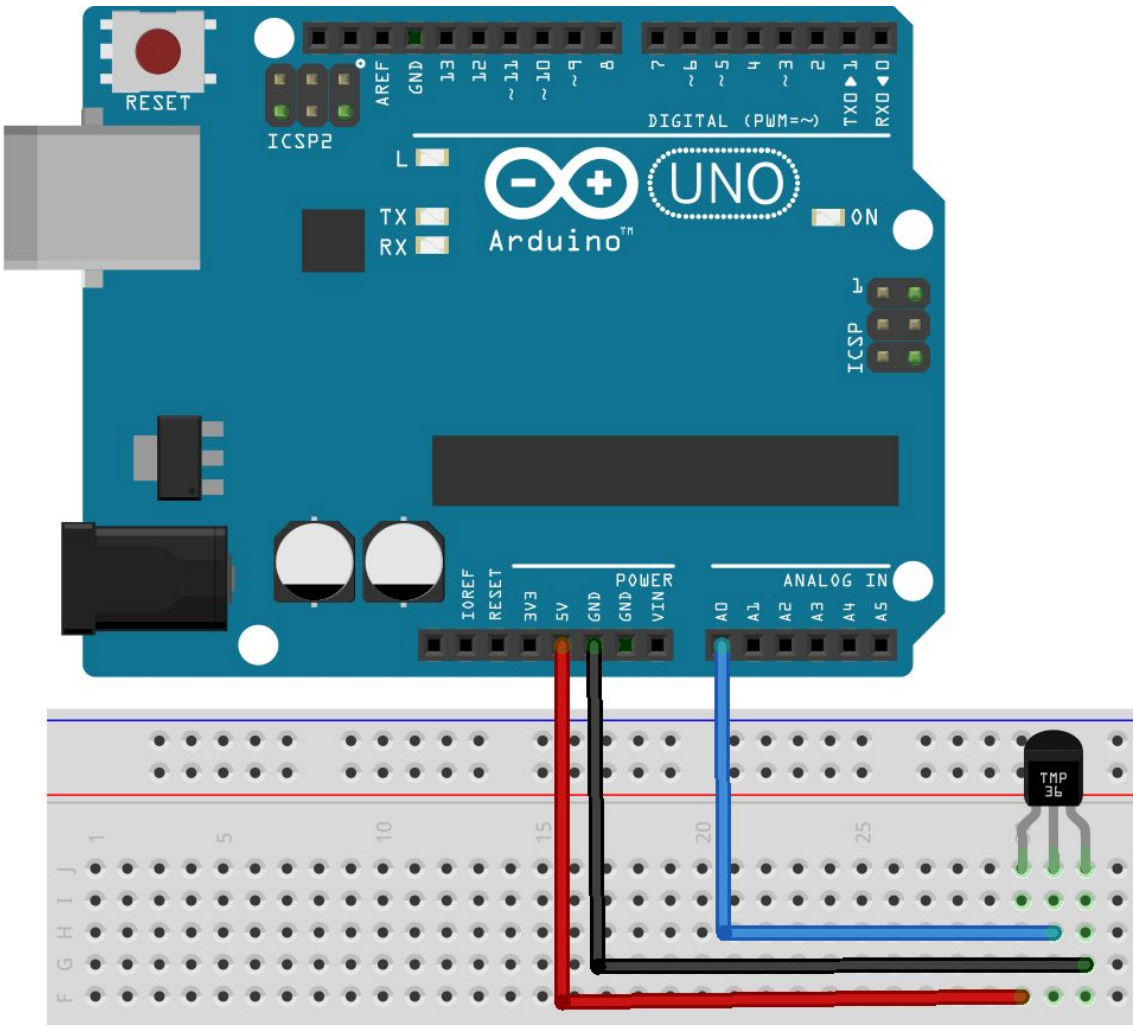
dove $\mathbf{V_q}$ ricordo essere il valore restituito dalla **`analogRead(pin)`**

rilevare temperature 1/3 sketch32

Utilizzando lo stesso circuito del precedente esperimento si realizzi lo sketch che rileva la temperatura letta dal sensore.

Componenti:

- Arduino
- TMP36



Prof. Michele Maffucci

rilevare temperature 2/3 sketch32

```
/* Prof. Michele Maffucci
   26.03.2014

   Rilevare la temperatura con
   sensore di temperatura
   TMP36

   Questo codice è di dominio pubblico

*/

int valore = 0;              // valore restituito dall'analogRead()
float temperatura = 0;       // variabile per conservare
                             // la temperatura rilevata
void setup()
{
  Serial.begin(9600);        // inizializzazione della porta seriale
}

void loop()
{

  valore = analogRead(0);    // valore letto dal sensore

  // calcola la proporzione
  // il valore restituito da analogRead() è un numero compreso tra
  // 0 e 1024 pertanto ciascuna unità vale 5/1024 = 4,88 mV
  // da cui valore misurato (volt) = val * 4,88 mV

  temperatura = (0.488*valore-50);
  Serial.println(temperatura);     // stampa sulla Serial monitor
                                   // il valore letto dal sensore
  delay(1000);                     // intervallo di 1 secondo
}
```

Aumento della precisione modo 1

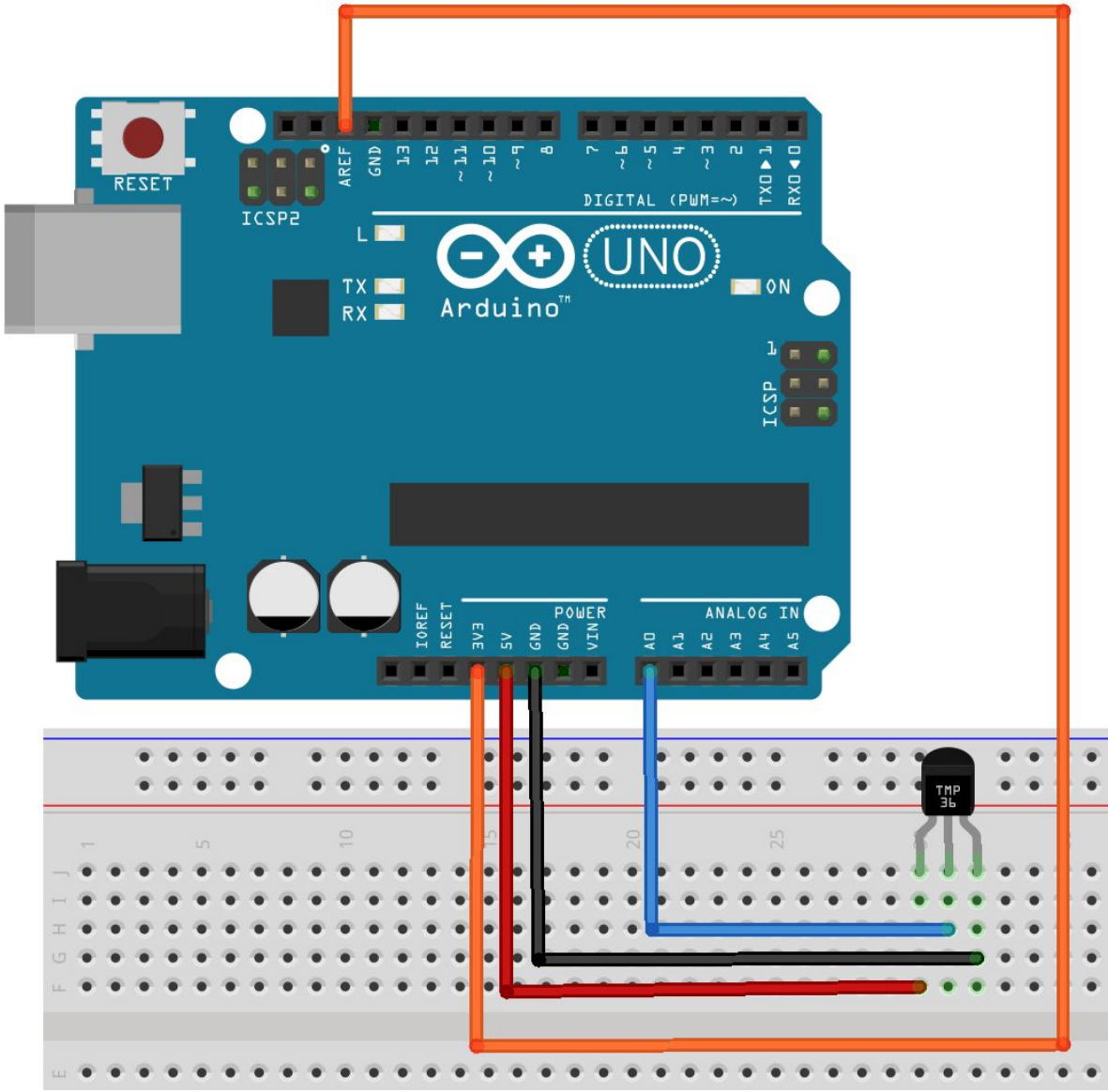
Per aumentare la precisione della temperatura rilevata si utilizza il pin AREF per indicare al convertitore analogico digitale di Arduino un valore più basso. Si prende come riferimento la tensione di 3,3V presente sulla scheda e si collega al pin AREF. Sostituendo questo valore nelle formule indicate in precedenza avremo:

$$\textbf{Temperatura} = 0{,}322 \times V_q - 50$$

dove V_q ricordo essere il valore restituito dalla **`analogRead(pin)`**

Componenti:

- Arduino
- TMP36



rilevare temperature 2/2 sketch33

```
/* Prof. Michele Maffucci
   26.03.2014

   Rilevare la temperatura con
   sensore di temperatura
   TMP36
   Aumentare la precisione di lettura
   usando il riferimento esterno di 3,3 V

   Questo codice è di dominio pubblico

*/

int valore = 0;              // valore restituito dall'analogRead()
float temperatura = 0;       // variabile per conservare
                             // la temperatura rilevata
void setup()
{
  Serial.begin(9600);        // inizializzazione della porta seriale
  analogReference(EXTERNAL); // viene indicato al convertitore AD che la tensione
                             // di riferimento non è più 5V ma quella ESTERNA
                             // di 3,3 V
}

void loop()
{

  valore = analogRead(0);    // valore letto dal sensore

  // calcola la proporzione
  // il valore restituito da analogRead() è un numero compreso tra
  // 0 e 1024 pertanto ciascuna unità vale 3,3/1024 = 0,322 mV
  // da cui valore misurato (volt) = val * 0,322 mV

  temperatura = (0.322*valore-50);
  Serial.println(temperatura);     // stampa sulla Serial monitor
                                   // il valore letto dal sensore
  delay(1000);                     // intervallo di 1 secondo
}
```

analogReference(EXTERNAL)

Indichiamo al microcontrollore viene utilizzata una tensione di riferimento esterna, in questo caso 3,3 V, tensione fornita direttamente su uno dei pin di alimentazione di Arduino.

Attenzione che nel caso si intendesse utilizzare nuovamente come riferimento di lettura il valore di default di 5V bisognerà impostare:

analogReference(DEFAULT)

Per approfondimenti seguire il link.

Aumento della precisione modo 2

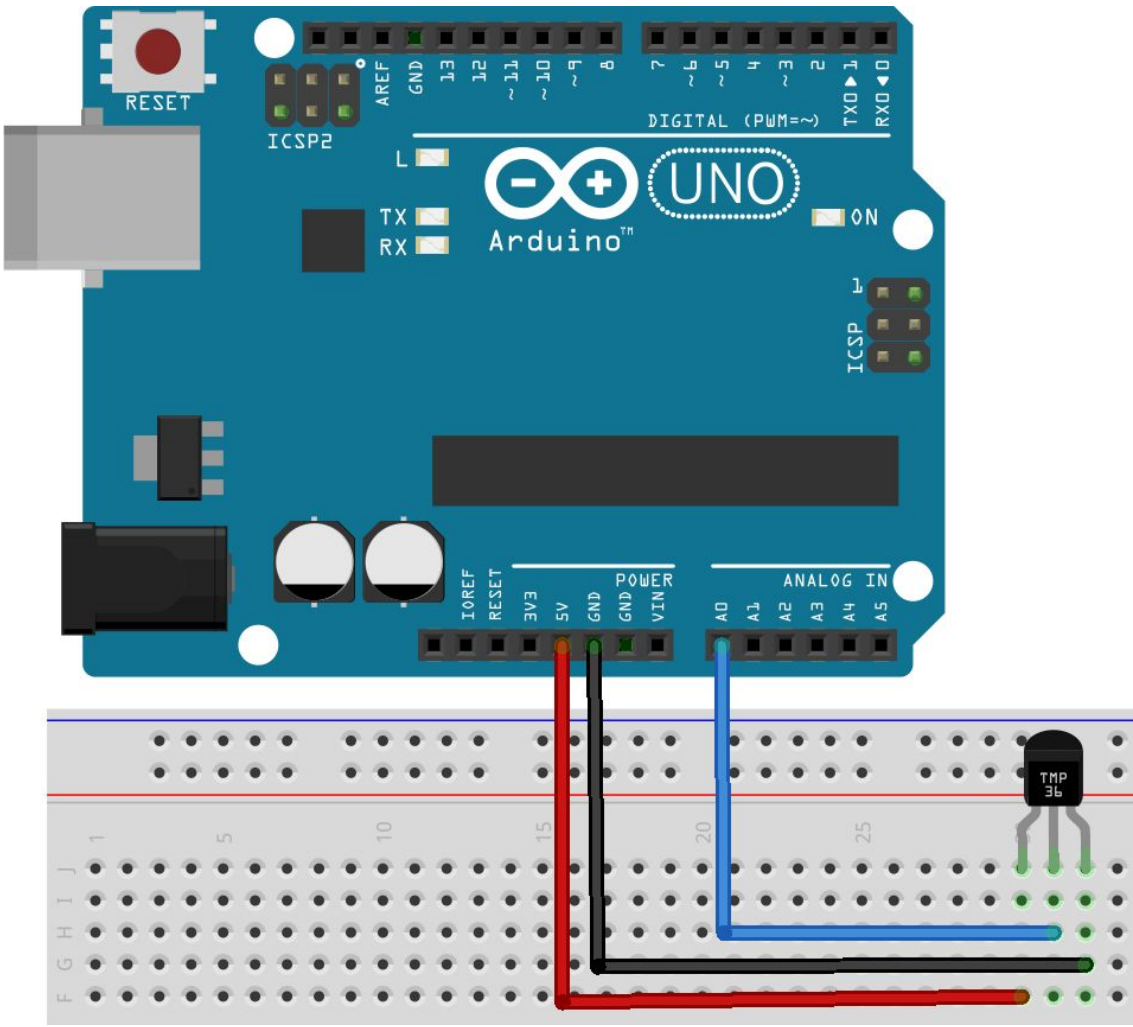
Per aumentare la precisione della temperatura rilevata si utilizza la tensione di riferimento interna di 1,1 V. Sostituendo questo valore nelle formule indicate in precedenza avremo:

$$\textbf{Temperatura} = 0{,}107 \times \mathbf{V_q} - 50$$

dove $\mathbf{V_q}$ ricordo essere il valore restituito dalla **`analogRead(pin)`**

Componenti:

- Arduino
- TMP36



rilevare temperature 2/2 sketch34

```
/* Prof. Michele Maffucci
   26.03.2014

   Rilevare la temperatura con
   sensore di temperatura
   TMP36
   Aumentare la precisione di lettura
   usando il riferimento interno di 1,1 V

   Questo codice è di dominio pubblico

*/

int valore = 0;              // valore restituito dall'analogRead()
float temperatura = 0;       // variabile per conservare
                             // la temperatura rilevata
void setup()
{
  Serial.begin(9600);        // inizializzazione della porta seriale
  analogReference(INTERNAL); // viene indicato al convertitore AD che la tensione
                             // di riferimento non è più 5V ma quella interna
                             // ma il valore INTERNO di 1,1 V
}

void loop()
{

  valore = analogRead(0);    // valore letto dal sensore

  // calcola la proporzione
  // il valore restituito da analogRead() è un numero compreso tra
  // 0 e 1024 pertanto ciascuna unità vale 1,1/1024 = 0,107 mV
  // da cui valore misurato (volt) = val * 0,107 mV

  temperatura = (0.107*valore-50);
  Serial.println(temperatura);     // stampa sulla Serial monitor
                                   // il valore letto dal sensore
  delay(1000);                     // intervallo di 1 secondo
}
```

analogReference(INTERNAL)

Indichiamo al microcontrollore viene utilizzata una tensione di riferimento INTERNA, che ha valore 1,1 V.

Attenzione che nel caso si intendesse utilizzare nuovamente come riferimento di lettura il valore di default di 5V bisognerà impostare:

analogReference(DEFAULT)

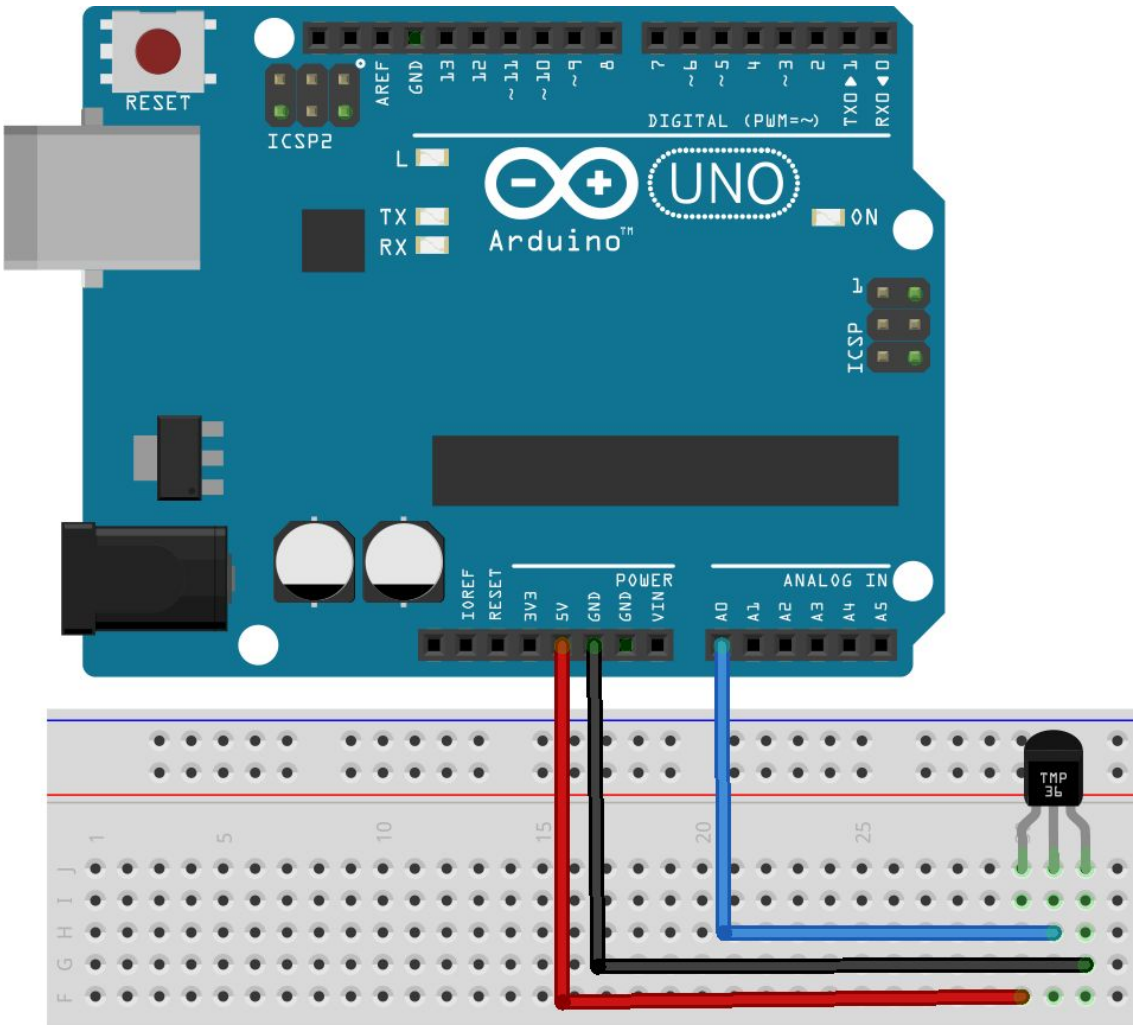
Per approfondimenti seguire il link.

Aumento della precisione modo 3

Per aumentare la precisione della temperatura rilevata si utilizza la tensione di riferimento interna di 1,1 V e si fa una media ta 8 valori letti.

Componenti:

- Arduino
- TMP36



rilevare temperature | 2/3 | sketch35

```
/* Prof. Michele Maffucci
   26.03.2014

   Rilevare la temperatura con
   sensore di temperatura
   TMP36
   Aumentare la precisione di lettura
   usando il riferimento interno di 1,1 V
   e facendo una media tra 8 rilevazioni

   Questo codice è di dominio pubblico

*/

int valore = 0;                  // valore restituito dall'analogRead()
float temperatura = 0;           // variabile per conservare
                                 // la temperatura rilevata

int misure[8];                   // array in cui memorizzare 8 valori
int i;

void setup()
{
  Serial.begin(9600);            // inizializzazione della porta seriale
  analogReference(INTERNAL);     // viene indicato al convertitore AD che la tensione
                                 // di riferimento non è più 5V ma quella interna
                                 // ma il valore INTERNO di 1,1 V
}
```

continua...

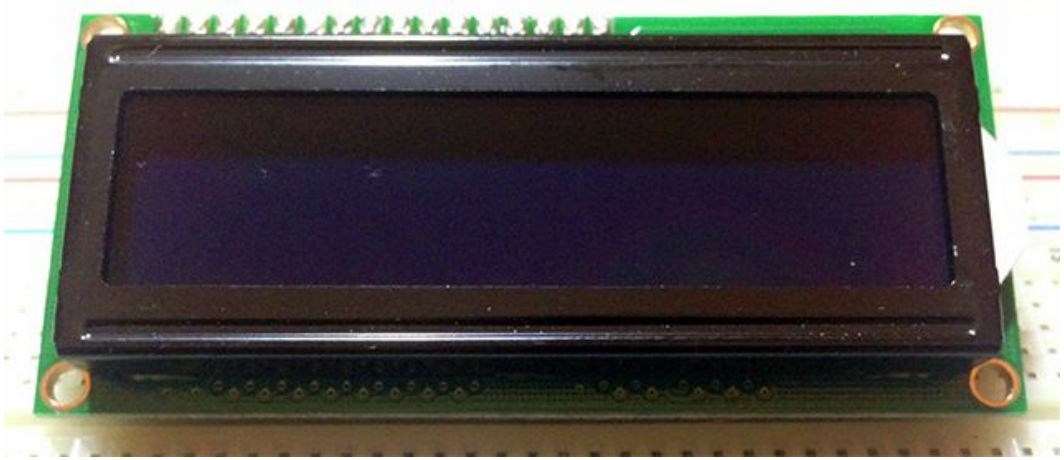
rilevare temperature | 3/3 | sketch35

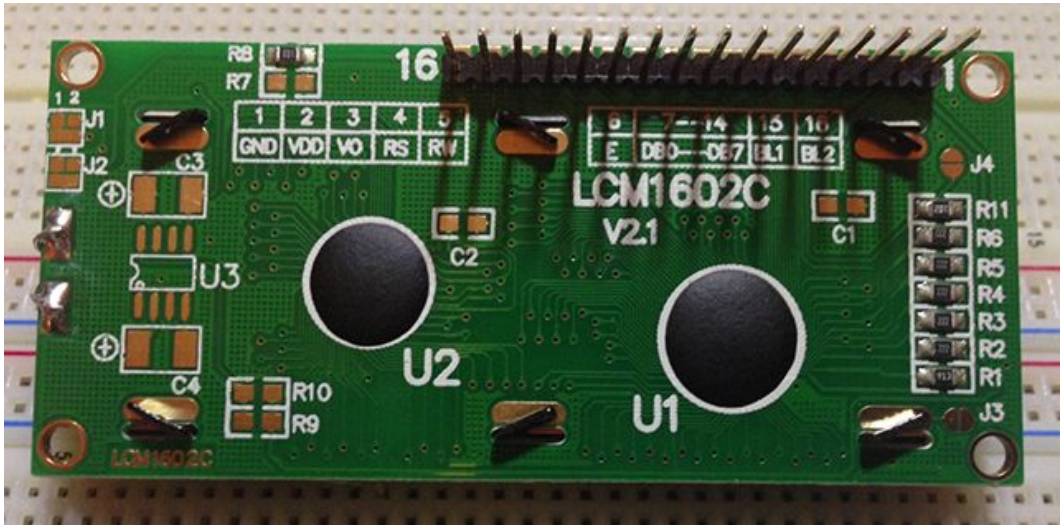
```
void loop()
{

  for(int i=0;i<=7;i++) {                    // legge 8 misure di temperatura

  valore = analogRead(0);                    // valore letto dal sensore

  // calcola la proporzione
  // il valore restituito da analogRead() è un numero compreso tra
  // 0 e 1024 pertanto ciascuna unità vale 1,1/1024 = 0,107 mV
  // da cui valore misurato (volt) = val * 0,107 mV

  misure[i] = (0.107*valore-50);             // memorizzazione nella posizione i-esima
                                             // della temperatura
  temperatura = temperatura + misure[i];     // somma al valore letto quello precedente
}

  temperatura = temperatura/8.0;             // media dei valori letti
  Serial.println(temperatura);               // stampa sulla Serial monitor
                                             // il valore letto dal sensore
  delay(1000);                               // intervallo di 1 secondo
}
```

Pilotare un display LCD

Prof. Michele Maffucci

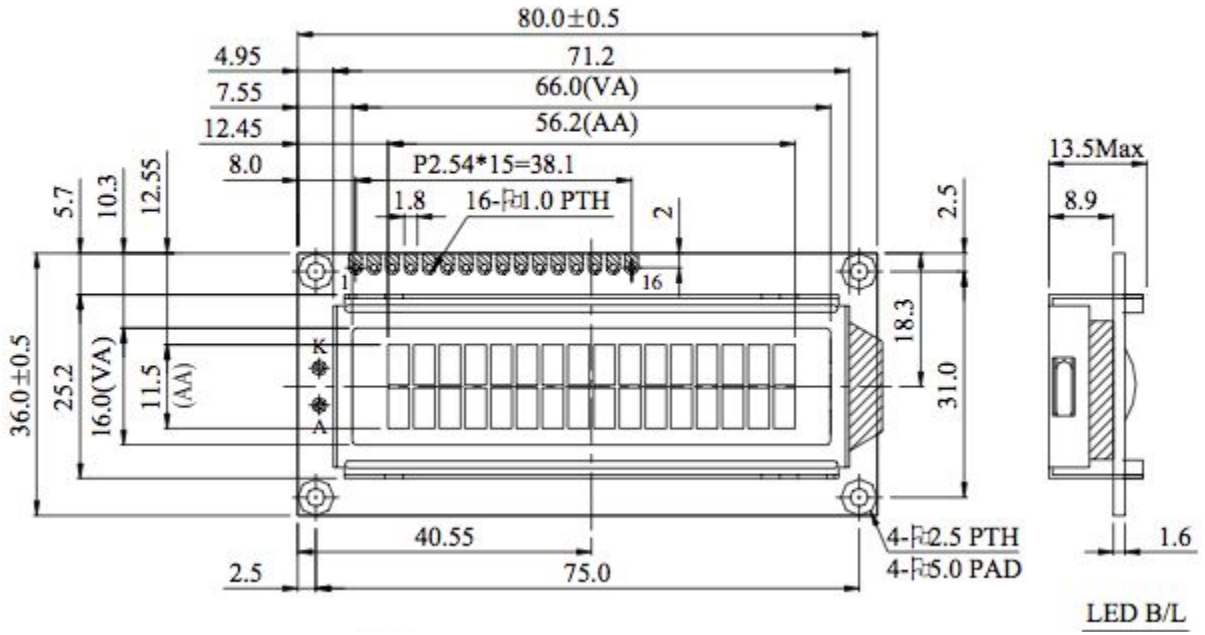
Display LCD 16×2 (16 colonne e 2 righe) compatibile con i driver dell'Hitachi HD44780





Per poter pilotare il display è indispensabile utilizzare la libreria **LiquidCrystal.h** che permette di comunicare in modalità 4 bit o 8 bit, questo vuol dire che per la trasmissione dati possono essere utilizzate 4 o 8 linee di controllo a queste si aggiungono le due linee di controllo: Register Select (RS) e Enable (E) e alla linea opzionale Read/Write (RW).
In questa lezione si utilizza una modalità a 4 bit, ciò comporta l'utilizzo di 6 uscite digitali sulla scheda Arduino.

Dall'immagine tratta dal datasheet, notate che il display è dotato di 16 pin e la numerazione parte da sinistra.



Nella tabella allegata le funzioni di ogni piedino:

Pin No.	Symbol	Level	Description
1	V_{SS}	0V	Ground
2	V_{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	A	—	LED +
16	K	—	LED—

La piedinatura di questi display è comune alla maggior parte degli LCD 16 x 2 standard Hitachi HD44780, ma potreste trovare qualche variazione tipicamente accade per i pin 15 e 16 che potrebbero essere al posto dei pin 1 e 2 (non è il caso di questo display). Come evidenziato nelle precedenti slide il display reca sul lato piste, una legenda con il riferimento piedino > funzione.

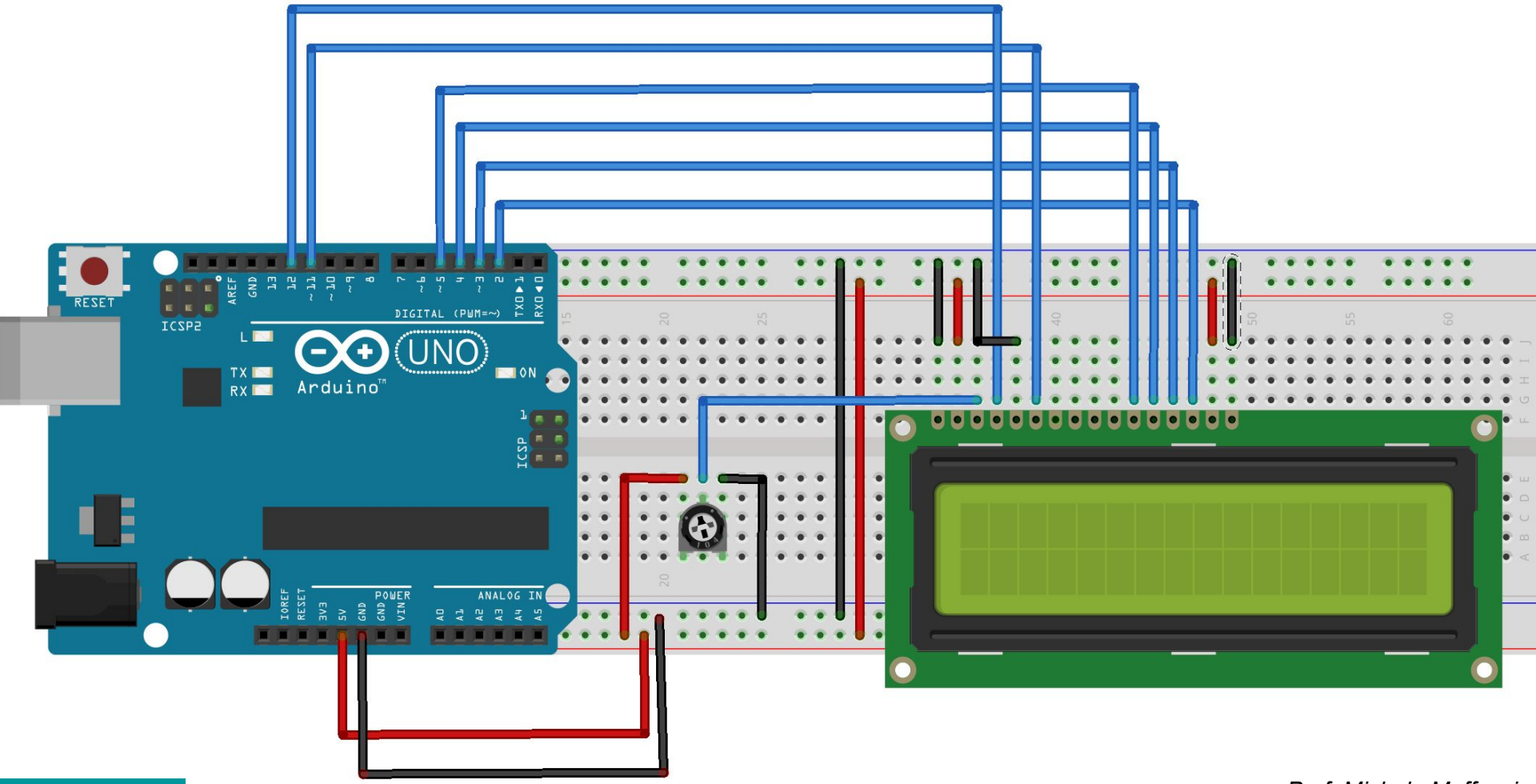
Funzione dei piedini:

- **Pin 1**: Vss – collegato al GND
- **Pin 2**: VDD – collegato a +5V
- **Pin 3**: V0 – controllo del contrasto dei caratteri. In genere viene collegato ad un potenziometro o trimmer in configurazione partitore di tensione in modo che possiate applicare sul Pin 3 una tensione che varia da 0 a +5V e al variare della tensione varia il contrasto.
- **Pin 4**: RS segnale di selezione registro – per selezionare il registro nel quale registrare ciò che appare sul display oppure selezionare il registro di funzionamento in cui viene memorizzata di volta in volta l'istruzione da eseguire per il funzionamento dell'LCD
- **Pin 5**: segnale Read/Write – per selezionare la modalità di funzionamento: lettura/scrittura – collegato a GND
- **Pin 6**: segnale Enable (E) – per abilitare la scrittura nei registri
- **Pin 7 al Pin 14**: linee dati che si inviano o si ricevono dai registri del display. Un valore HIGH (H) indica scrittura (WRITE) del bit nel registro del display, un valore LOW (L) indica un valore letto (READ) da un registro.
- **Pin 15**: A (Anodo) – piedino a cui collegare una tensione positiva (nel caso del display descritto +4, 2V) che serve per la retroilluminazione del display.
- **Pin 16**: K (Catodo) – piedino da collegare a GND per consentire la retroilluminazione.

scrivere un testo | 1/3 | sketch36

Realizzare il circuito riportato in figura ed eseguire lo sketch corrispondente che permette di scrivere un testo sul display e un contatore di secondi.

Componenti: Arduino, Display 16x2, Trimmer da 10KOhm



Prof. Michele Maffucci

scrivere un testo | 2/3 | sketch36

```
/* Prof. Michele Maffucci
   26.03.2014

   Uso del display LCD 16x2 standard Hitachi HD44780

   Questo codice è di dominio pubblico

  Circuito:
 * pin RS collegato al pin digitale 12
 * pin E (Enable) collegato al pin digitale 11
 * pin D4 collegato al pin digitale 5
 * pin D5 collegato al pin digitale 4
 * pin D6 collegato al pin digitale 3
 * pin D7 collegato al pin digitale 2
 * pin R/W collegato al GND
 * pin 1 e pin 4 collegati a GND
 * pin 2 collegato a +Vcc
 * centrale del potenziometro/trimmer da 10 KOhm collegato al pin 3 del'LCD
 * pin SX potenziometro/trimmer collegato a +Vcc
 * pin DX potenziometro/trimmer collegato a GND
 * i pin SX e DX del potenziometro/trimmer possono essere interscambiati
*/

// includere la libreria:
#include <LiquidCrystal.h>

/*
   Viene creata l'istanza dell'oggetto LiquidCrystal chiamata lcd in cui
   sono indicati i pin dell'LCD collegati alle uscite digitali di Arduino
*/

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

continua...

scrivere un testo | 3/3 | sketch36

```
void setup() {
   //impostiamo il numero di colonne ed il numero di righe di lcd
  lcd.begin(16, 2);
  // Visualizzo il messaggio sul display
  lcd.print("Salve mondo!");
}

void loop() {
  // posiziona il cursore in colonna 0 e linea 1
  // (nota: la linea 1 e la seconda linea, poichè si conta incominciando da 0):
  lcd.setCursor(0, 1);
  // stampa il numero di secondi dall'ultimo reset
  lcd.print(millis()/1000);
}
```

continua...

Per approfondimenti sull'uso del display si rimanda al sito Arduino, sezione **Tutorial: LiquidCrystal Library**:

- Blink: controllo del cursore.
- Cursor: controllo cursore sottolineato.
- Display: far lampeggiare un testo.
- TextDirection: controllare la direzione in cui il testo viene scritto.
- Scroll: spostamento del testo da destra a sinistra e viceversa.
- Serial input: scrittura del testo inviato dalla serial input.
- SetCursor: impostare la posizione del cursore.
- Autoscroll: spostamento di un testo da sinistra a destra e viceversa.

continua...

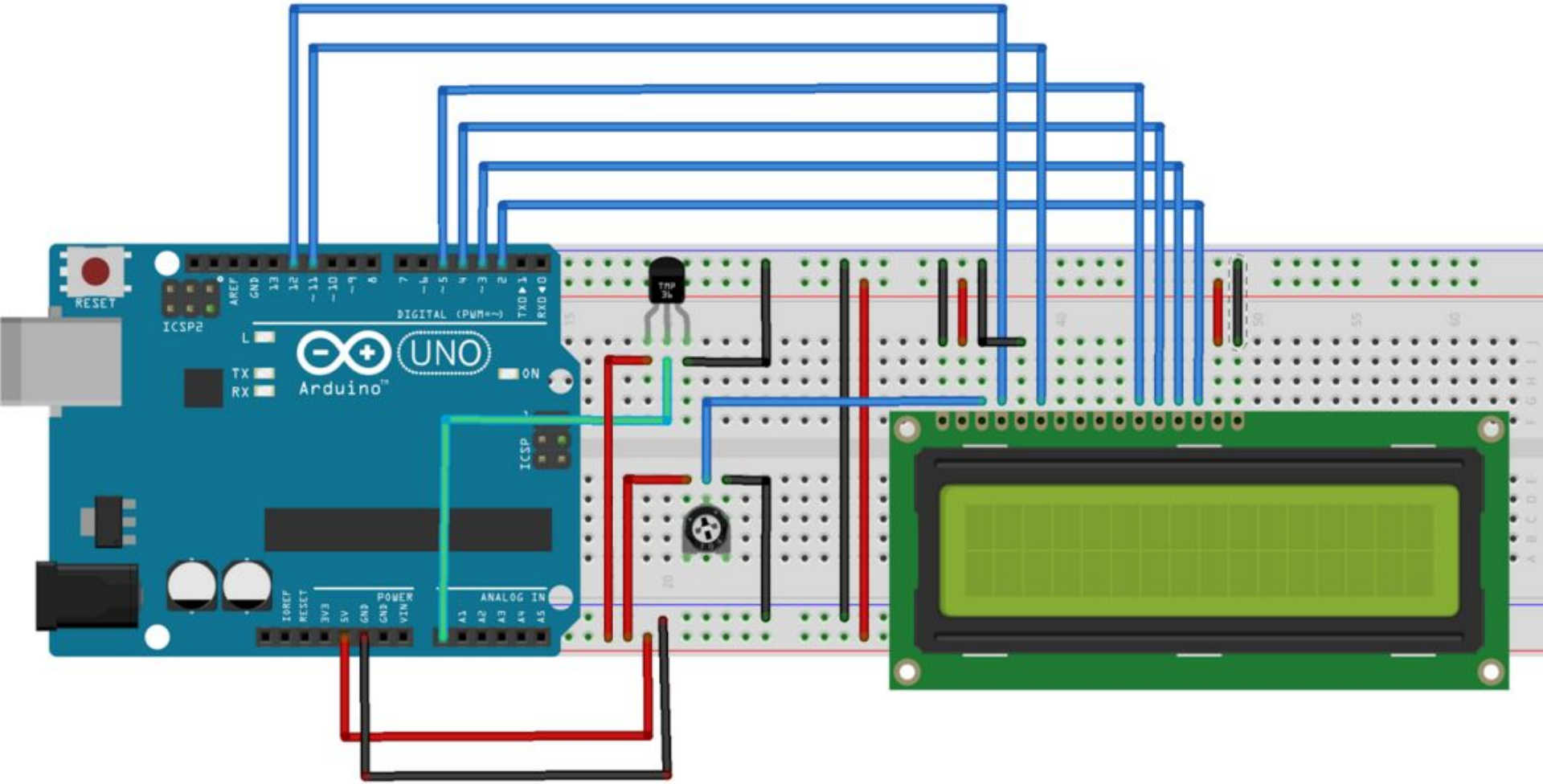
visualizzare la temperatura ambiente su LCD

1/3

sketch37

Visualizzare su display la temperatura ambiente.

Componenti: Arduino, Display 16x2, Trimmer da 10KOhm, TMP36



visualizzare la temperatura ambiente su LCD 2/3 sketch37

```
/* Prof. Michele Maffucci
   26.03.2014

   Visualizare la temperatura rilevata con un TMP36
   su un display LCD 16x2 standard Hitachi HD44780

   Questo codice è di dominio pubblico

  Circuito:
 * pin RS collegato al pin digitale 12
 * pin E (Enable) collegato al pin digitale 11
 * pin D4 collegato al pin digitale 5
 * pin D5 collegato al pin digitale 4
 * pin D6 collegato al pin digitale 3
 * pin D7 collegato al pin digitale 2
 * pin R/W collegato al GND
 * pin 1 e pin 4 collegati a GND
 * pin 2 collegato a +Vcc
 * centrale del potenziometro/trimmer da 10 KOhm collegato al pin 3 del'LCD
 * pin SX potenziometro/trimmer collegato a +Vcc
 * pin DX potenziometro/trimmer collegato a GND
 * i pin SX e DX del potenziometro/trimmer possono essere interscambiati
*/

// includere la libreria:
#include <LiquidCrystal.h>

/*
   Viene creata l'istanza dell'oggetto LiquidCrystal chiamata lcd in cui
   sono indicati i pin dell'LCD collegati alle uscite digitali di Arduino
*/

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

continua...

visualizzare la temperatura ambiente su LCD

3/3

sketch37

```
int valore = 0;                  // valore restituito dall'analogRead()
float temperatura = 0;           // variabile per conservare
                                 // la temperatura rilevata

float misure[8];                   // array in cui memorizzare 8 valori

void setup() {
  lcd.begin(16, 2);           //impostiamo il numero di colonne ed il numero di righe di lcd
  analogReference(INTERNAL);  // viene indicato al convertitore AD che la tensione
                              // di riferimento non è più 5V ma quella interna
                              // ma il valore INTERNO di 1,1 V
}

void loop() {
  // stampa della prima riga sull'LCD
  lcd.print("Temperatura...");
  // posiziona il cursore in colonna 0 e linea 1
  // (nota: la linea 1 e la seconda linea, poichè si conta incominciando da 0):
  lcd.setCursor(0, 1);

  for(int i=0;i<=7;i++) {                    // legge 8 misure di temperatura

  valore = analogRead(0);                    // valore letto dal sensore

  // calcola la proporzione
  // il valore restituito da analogRead() è un numero compreso tra
  // 0 e 1024 pertanto ciascuna unità vale 1,1/1024 = 0,107 mV
  // da cui valore misurato (volt) = val * 0,107 mV

  misure[i] = (0.107*valore-50);             // memorizzazione nella posizione i-esima
                                             // della temperatura
  temperatura = temperatura + misure[i];     // somma al valore letto quello precedente
}

  temperatura = temperatura/8.0;             // media dei valori letti
  lcd.print(temperatura);                    // stampa la temperatura sull'LCD
  lcd.print(" Celsius ");                    // stampa la la parola "Celsius" dopo il valore
  delay(1000);                               // intervallo di 1 secondo
  lcd.clear();                               // cancella lo schermo
  temperatura=0;                             // inizializzazione della variabile per
                                             // il nuovo ciclo di lettura
}
```

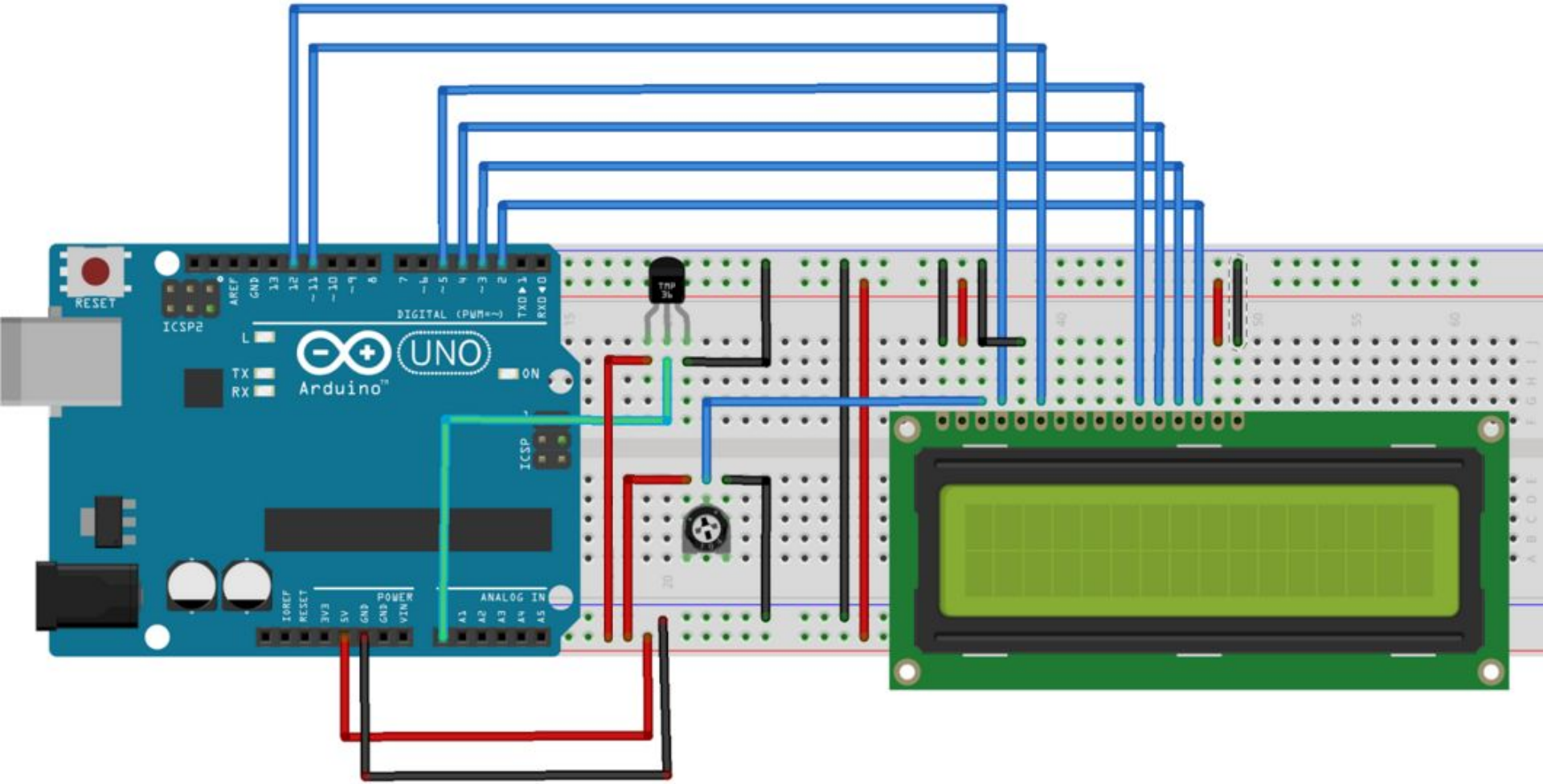
visualizzare la temperatura con allarme

1/3

sketch38

Visualizzare su display la temperatura ambiente. Se la temperatura supera un valore massimo ed un valore minimo fissati viene visualizzato il messaggio di allarme

Componenti: Arduino, Display 16x2, Trimmer da 10KOhm, TMP36



Prof. Michele Maffucci

visualizzare la temperatura ambiente su LCD 2/3 sketch38

```
/* Prof. Michele Maffucci
   26.03.2014

   Visualizare la temperatura rilevata con un TMP36
   con allarme di temperatura massima e minima
   su un display LCD 16x2 standard Hitachi HD44780

   Questo codice è di dominio pubblico

  Circuito:
 * pin RS collegato al pin digitale 12
 * pin E (Enable) collegato al pin digitale 11
 * pin D4 collegato al pin digitale 5
 * pin D5 collegato al pin digitale 4
 * pin D6 collegato al pin digitale 3
 * pin D7 collegato al pin digitale 2
 * pin R/W collegato al GND
 * pin 1 e pin 4 collegati a GND
 * pin 2 collegato a +Vcc
 * centrale del potenziometro/trimmer da 10 KOhm collegato al pin 3 del'LCD
 * pin SX potenziometro/trimmer collegato a +Vcc
 * pin DX potenziometro/trimmer collegato a GND
 * i pin SX e DX del potenziometro/trimmer possono essere interscambiati
*/

// includere la libreria:
#include <LiquidCrystal.h>

/*
   Viene creata l'istanza dell'oggetto LiquidCrystal chiamata lcd in cui
   sono indicati i pin dell'LCD collegati alle uscite digitali di Arduino
*/

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

continua...

visualizzare la temperatura ambiente su LCD 2/3 sketch38

```
int valore = 0;                  // valore restituito dall'analogRead()
float temperatura = 0;           // variabile per conservare
                                 // la temperatura rilevata

float misure[8];                   // array in cui memorizzare 8 valori

float tempMax = 25;              // valore massimo di allarme della temperatura
float tempMin = 20;              // valore minimo di allarme della temperatura

void setup() {
  lcd.begin(16, 2);           //impostiamo il numero di colonne ed il numero di righe di lcd
  analogReference(INTERNAL);  // viene indicato al convertitore AD che la tensione
                              // di riferimento non è più 5V ma quella interna
                              // ma il valore INTERNO di 1,1 V
}

void loop() {
  // stampa della prima riga sull'LCD
  lcd.print("Temperatura...");
  // posiziona il cursore in colonna 0 e linea 1
  // (nota: la linea 1 e la seconda linea, poichè si conta incominciando da 0):
  lcd.setCursor(0, 1);

  for(int i=0;i<=7;i++) {                    // legge 8 misure di temperatura

  valore = analogRead(0);                    // valore letto dal sensore

  // calcola la proporzione
  // il valore restituito da analogRead() è un numero compreso tra
  // 0 e 1024 pertanto ciascuna unità vale 1,1/1024 = 0,107 mV
  // da cui valore misurato (volt) = val * 0,107 mV

  misure[i] = (0.107*valore-50);             // memorizzazione nella posizione i-esima
                                             // della temperatura
  temperatura = temperatura + misure[i];     // somma al valore letto quello precedente
}

  temperatura = temperatura/8.0;             // media dei valori letti
```

continua...

visualizzare la temperatura ambiente su LCD 2/3 sketch38

```
  // allarme se si supera la temperatura massima
  if (temperatura > tempMax) {
    allarmeMax(temperatura);
  }

  // allarme se si scende sotto la temperatura minima
  if (temperatura < tempMin) {
    allarmeMin(temperatura);
  }

  // nessun allarme se si è nell'intervallo
  if (temperatura > tempMin && temperatura < tempMax) {
    stampaTemperatura(temperatura);             // stampa la temperatura
  }
  temperatura=0;                                // inizializzazione della variabile per
                                                // il nuovo ciclo di lettura
}

void stampaTemperatura(float temperatura){
  lcd.setCursor(0, 1);        // posiziona il cursore in colonna 0 riga 1
  lcd.print(temperatura);     // stampa la temperatura sull'LCD
  lcd.print((char)223);       // stampa il carattere °
  lcd.print("C");             // stampa il carattere C
  delay(1000);                // intervallo di 1 secondo
  lcd.clear();                // cancella lo schermo
}

void allarmeMax(float temperatura){
  lcd.clear();                    // cancella lo schermo
  lcd.setCursor(0, 0);            // posiziona il cursore in colonna 0 riga 0
  lcd.print("T. alta!");          // allarme temperatura alta
  stampaTemperatura(temperatura); // stampa la temperatura
}

void allarmeMin(float temperatura){
  lcd.clear();                    // cancella lo schermo
  lcd.setCursor(0, 0);            // posiziona il cursore in colonna 0 riga 0
  lcd.print("T. bassa!");         // allarme temperatura alta
  stampaTemperatura(temperatura); // stampa la temperatura
}
```

Grazie

Prof. Michele Maffucci

www.maffucci.it
michele@maffucci.it

www.twitter.com/maffucci/
www.facebook.com/maffucci.it/
plus.google.com/+MicheleMaffucci/

it.linkedin.com/in/maffucci