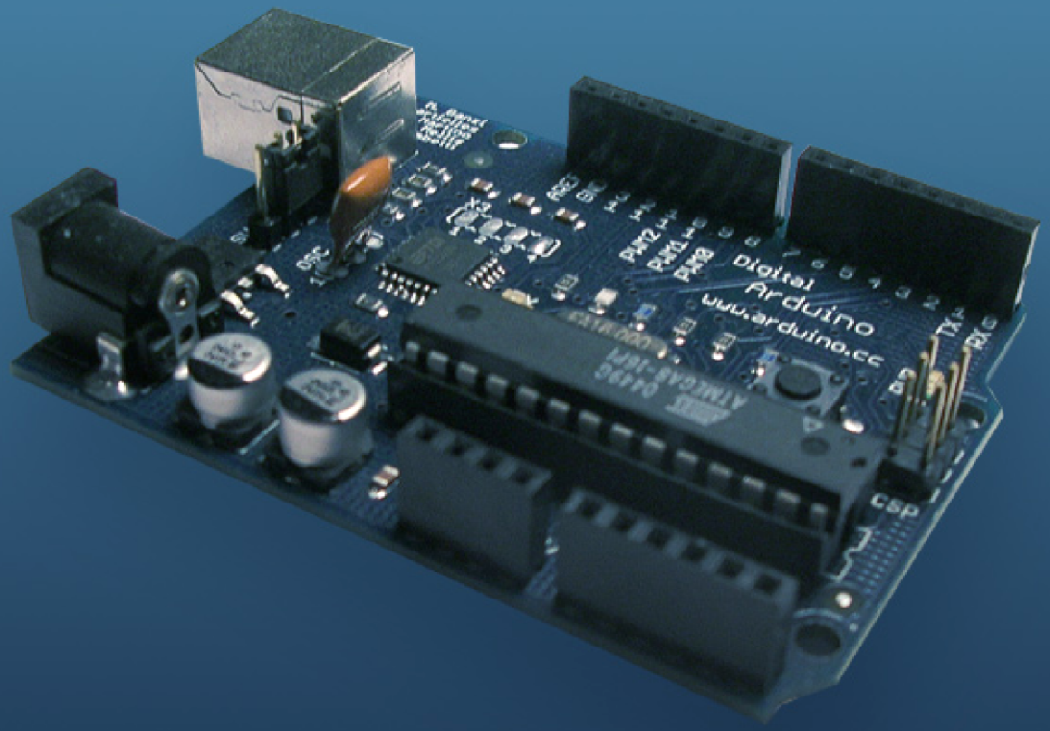
Arduino

Physical Computing I/O board



7^ parte : Acquisizione della temperatura con LM35
e visualizzazione su display LCD

Author: Ing. Sebastiano Giannitto (ITIS “M.BARTOLO” –PACHINO)

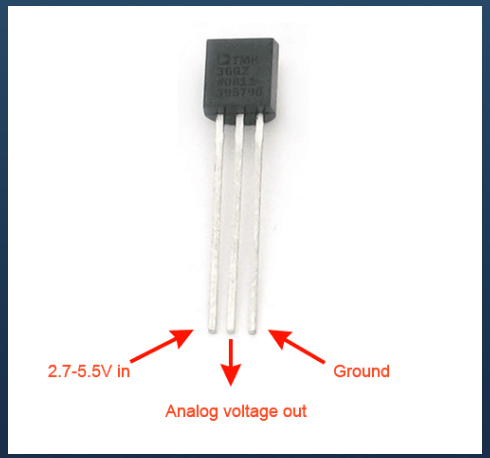
Esperienza n° 6

Lo scopo del progetto è realizzare un termometro digitale con **Arduino**. La temperatura verrà prelevata con il sensore di temperatura **LM35** e dovrà essere visualizzata su un **display LCD** e aggiornata ogni 1s.

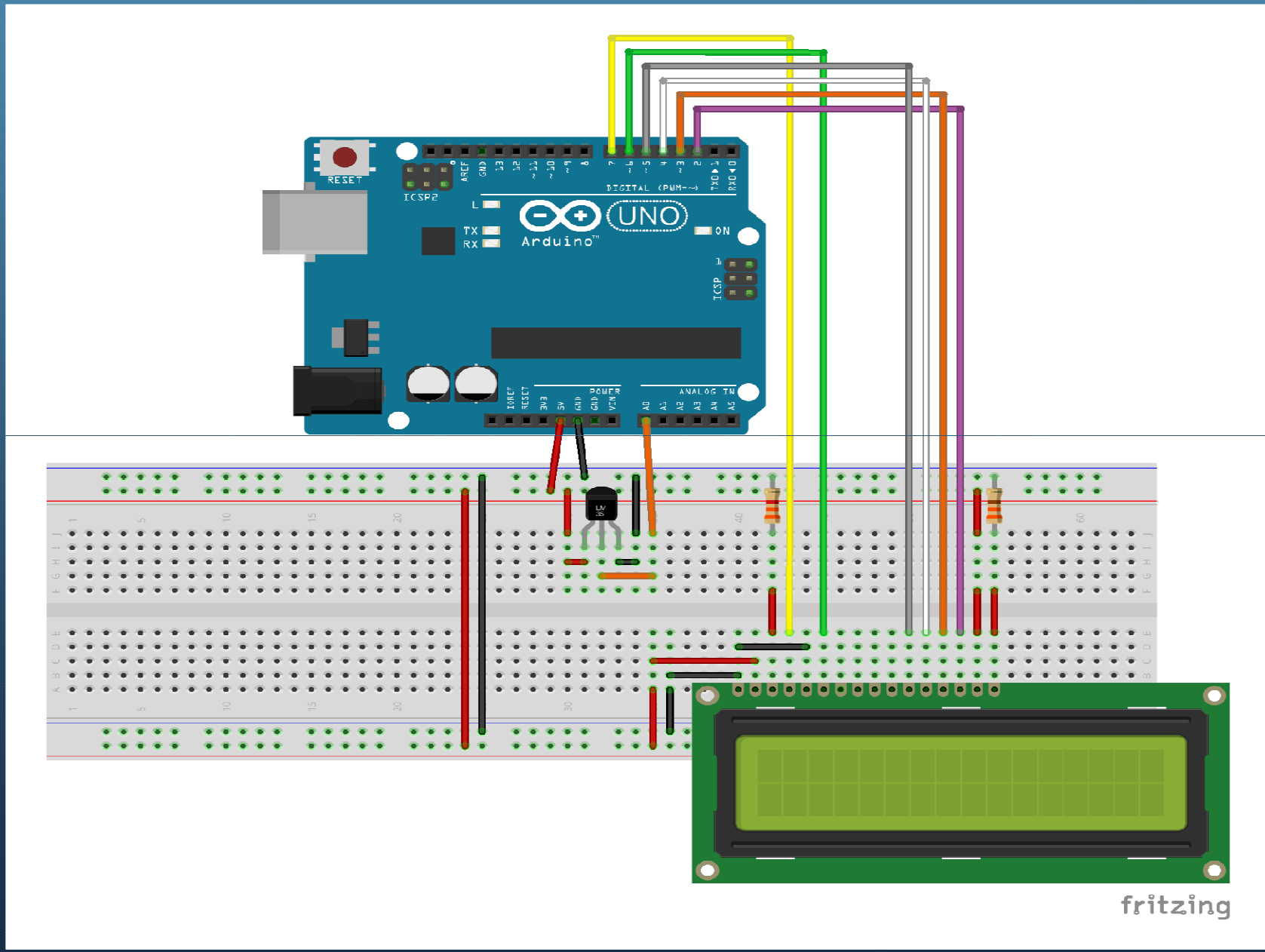
Necessario, oltre la solita breadboard e cavetteria di vario genere:

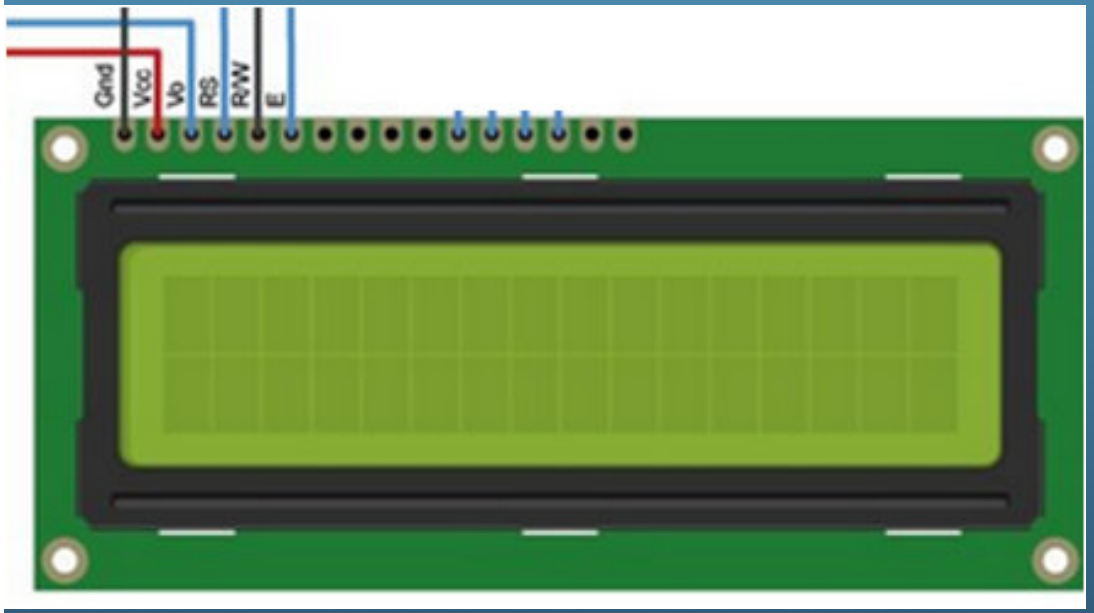
- **1 Display LCD 16x2 (HD44780)**
- 1 Sensore di temperatura **LM35**
- 1x Resistenza da 330ohm
- 1x Resistenza da 3,3kohm

Il sensore si presenta con 3 terminali: uno per l'alimentazione, uno di massa e uno per l'uscita della tensione proporzionale alla temperatura rilevata che è pari a 10 mV per ogni grado centigrado, ed è calibrato in gradi Cesius.



Schema e Circuito elettrico su breadboard



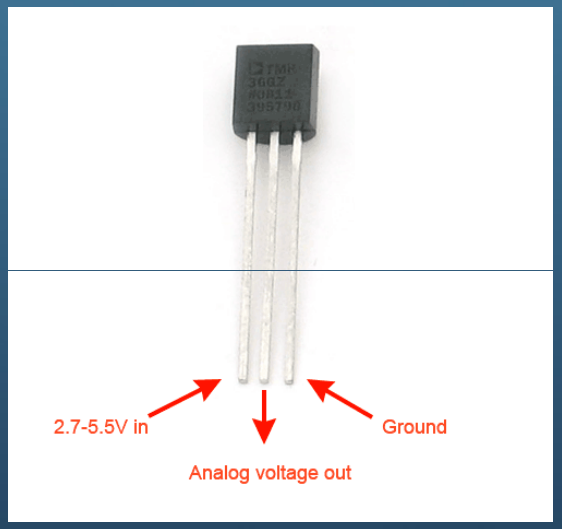


Collegamenti display LCD:

Pin n°	Collegato a
1	GND
2	5V
3	resistenza da 3,3 kΩ collegata a GND (vedi nota sotto in rosso)
4	pin 7 di Arduino
5	GND
6	pin 6 di Arduino
11	pin 5 di Arduino
12	pin 4 di Arduino
13	pin 3 di Arduino
14	pin 2 di Arduino
15	5V
16	resistenza da 330Ω collegata a GND

Fare attenzione!
Il pin Vo del display LCD regola il contrasto dei caratteri visualizzati nel display. Come riportato sopra, bisogna collegarlo ad una resistenza da 3,3 kΩ collegata a GND. Tuttavia in alcuni display potrebbe essere necessario collegare il pin Vo direttamente a GND.

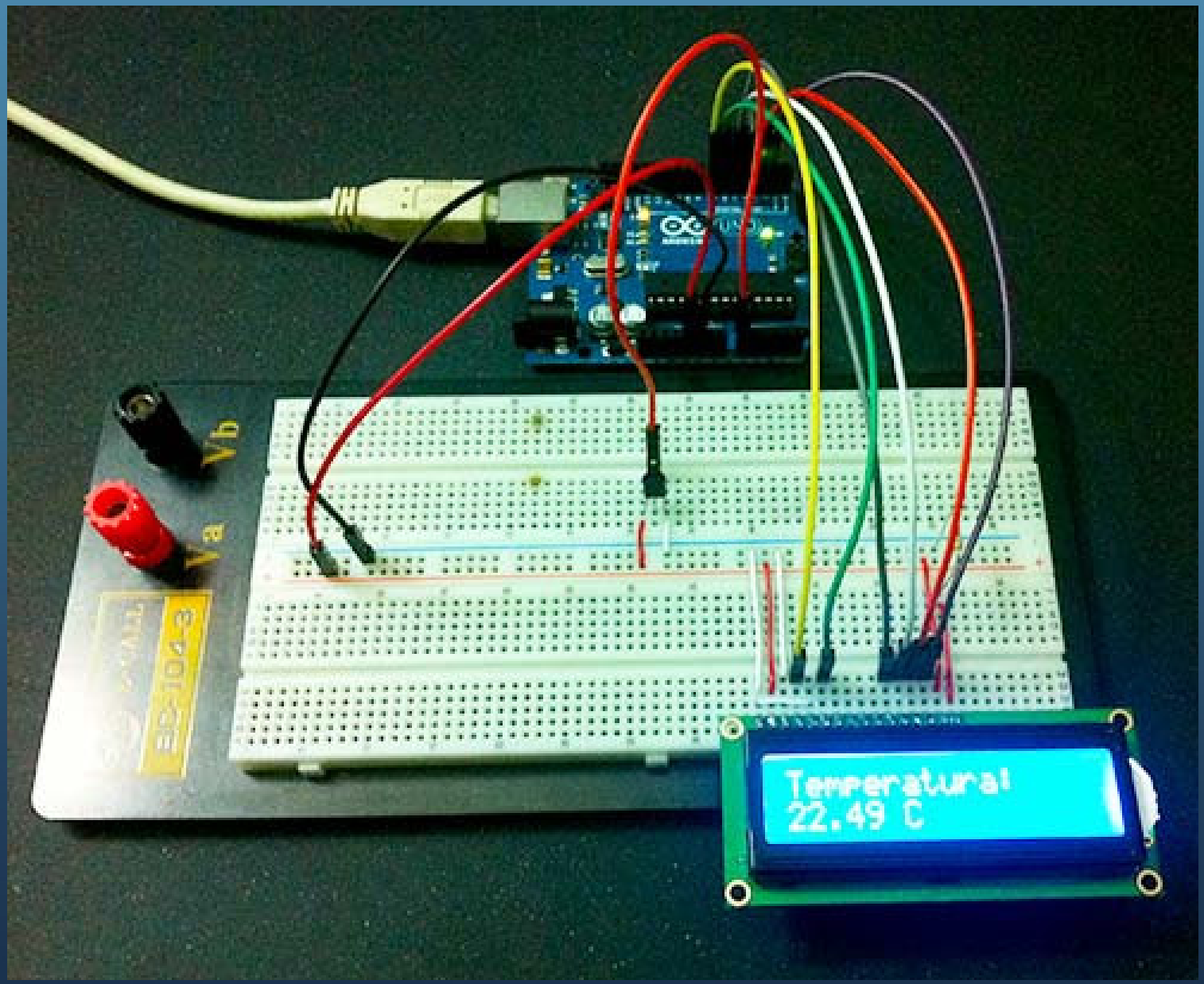
Dopo aver collegato il display LCD, è necessario collegare il sensore per prelevare la temperatura. Effettua i seguenti collegamenti riguardanti il sensore di temperatura LM35:
Collega il pin 1 a GND.
Collega il pin 2 al pin A0 di Arduino.
Collega il pin 3 a 5V.



Il sensore di temperatura **LM35** presenta un contenitore di tipo TO92 analogo a quello di un normale transistor, e necessita solo di un collegamento ad una porta ADC per la lettura del valore di tensione fornito dal sensore.

Le principali caratteristiche dell' LM35 sono:

Altezza	**5.2mm**
Profondità	**4.19mm**
Lunghezza	**5.2mm**
Package fornitore	**TO-92**
Funzione	**Sensore di temperatura**
Pin Count	**3**
Precisione	**±0.5℃**
Sensibilità	**10mV/℃**
Temperatura massima	**155℃**
Temperatura minima	**-50℃**
Tensione tipica di funzionamento	**4-30V**
Tipo uscita	**Analogica**


Vb
Va
Temperatura!
22.49 C

Il codice

```
#include <LiquidCrystal.h> //Libreria per pilotare il display LCD

#define pin_temp A0 //Pin di collegamento del piedino Vout del sensore di temperatura

float temp = 0; //Variabile in cui verrà memorizzata la temperatura rilevata

LiquidCrystal lcd(7, 6, 5, 4, 3, 2); //Inizializzazione della libreria con i pin del display LCD

void setup()
{
  lcd.begin(16, 2); //Impostazione del numero di colonne e righe del display LCD
  lcd.setCursor(0, 0); //Sposto il cursore sulla prima riga (riga 0) e sulla prima colonna
  lcd.print("Temperatura:"); //Stampo il messaggio 'Temperatura:' sulla prima riga

  /*Imposto Vref dell'ADC a 1,1V
  (per una maggiore precisione nel calcolo della temperatura)
  ------------------------------
  IMPORTANTE: Se utilizzi Arduino Mega sostituisci INTERNAL con INTERNAL1V1 */
  analogReference(INTERNAL);
}
```

Il codice

```
void loop()
{
  /*Calcolo la temperatura  ==============================================*/
  temp = 0;
  for (int i = 0; i < 5; i++) { //Esegue l'istruzione successiva 5 volte
    temp += (analogRead(pin_temp) / 9.31); //Calcola la temperatura e la somma alla variabile 'temp'
  }
  temp /= 5; //Calcola la media matematica dei  valori di temperatura
  /*=====================================================================*/
  /*Visualizzo la temperatura sul display LCD  ==========================================*/
  lcd.setCursor(0, 1); //Sposto il cursore sulla prima colonna e sulla seconda riga
  lcd.print(temp); //Stampo sul display LCD la temperatura
  lcd.print(" C"); //Stampo uno spazio e il carattere 'C' sul display
  /*=========================================*/
  delay(1000); //Ritardo di un secondo (può essere modificato)
}
```

Altro esempio

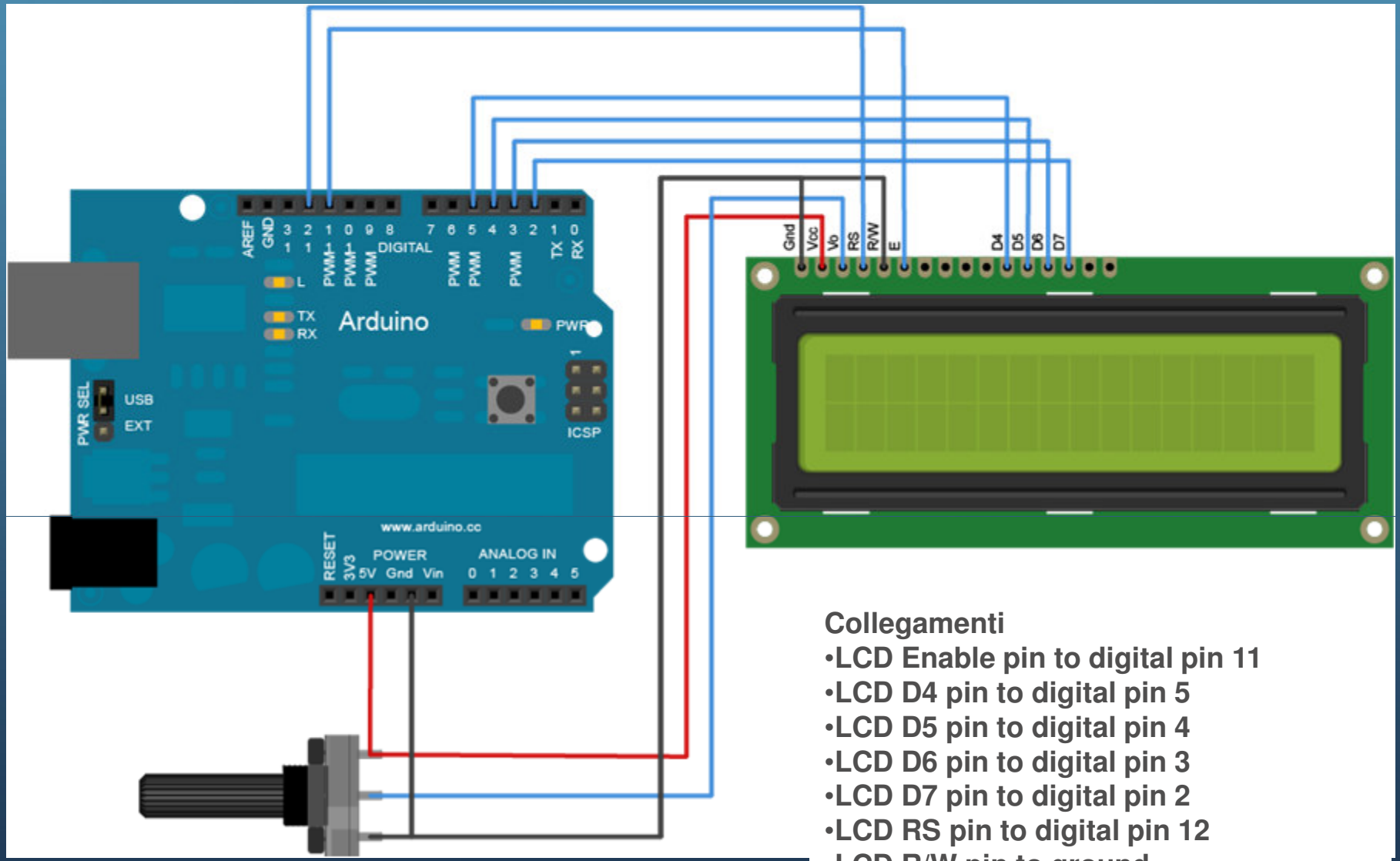
Stampare "Ciao Mondo!" allo schermo LCD e mostrare il tempo in secondi da quando Arduino è stato resettato.

I display LCD hanno un'interfaccia parallela, costituita dai seguenti pin:
Un pin (RS) **register select** che controlla dove nella memoria dell' LCD si sta scrivendo il dato. È possibile selezionare il **registro dei dati**, che detiene ciò che accade sullo schermo, o un **registro di istruzione**, che punta all'istruzione successiva.
A pin di lettura / scrittura (**R / W**) che seleziona la modalità di lettura o modalità di scrittura
Un pin **Enable** che consente la scrittura dei registri
8 pin dati (D0 -D7) . Gli stati di questi pin (alti o bassi) sono i bit che si stanno scrivendo o i valori che si stanno leggendo.

C'è anche un pin di constrasto (Vo) ,
piedini di alimentazione (+ 5V e GND) e un pin di retroilluminazione LED (Bklt + e BKlt -) che è possibile utilizzare per alimentare il display LCD, controllare il contrasto del display, e accendere e spegnere il LED retroilluminazione, rispettivamente.

Il processo di controllo del display consiste nel mettere i dati che formano l'immagine di ciò che si desidera visualizzare nei registro dati, poi mettendo le istruzioni nel registro delle istruzioni.

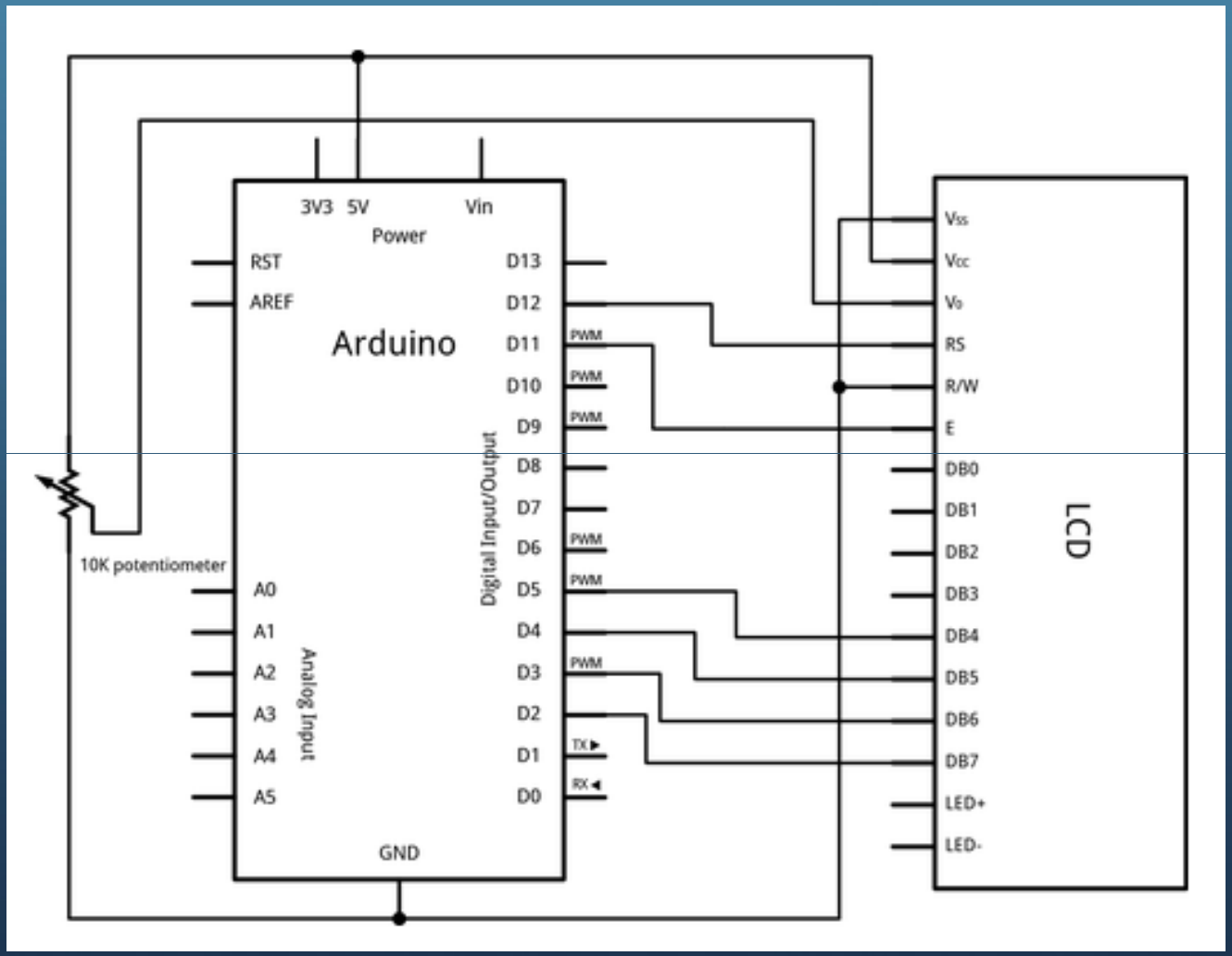
La **Biblioteca LiquidCrystal** semplifica questo per voi in modo che non è necessario conoscere le istruzioni di basso livello.
I compatibili Hitachi LCD possono essere controllati in due modi: 4 bit o 8 bit. La modalità a 4 bit richiede sette pin I / O del Arduino, mentre la modalità a 8 bit richiede 11 pin.
Per visualizzare il testo sullo schermo, si può fare quasi tutto in modalità a 4 bit.



Potenziometro da 10kΩ

Collegamenti

- **LCD Enable pin to digital pin 11**
- **LCD D4 pin to digital pin 5**
- **LCD D5 pin to digital pin 4**
- **LCD D6 pin to digital pin 3**
- **LCD D7 pin to digital pin 2**
- **LCD RS pin to digital pin 12**
- **LCD R/W pin to ground**
- **10K resistor:**
- **ends to +5V and ground**
- **wiper to LCD VO pin (pin 3)**


3V3
5V
Vin
Power
RST
AREF
Arduino
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
PWM
PWM
PWM
PWM
PWM
PWM
TX
RX
Digital Input/Output
10K potentiometer
A0
A1
A2
A3
A4
A5
Analog Input
GND
Vss
Vcc
Vo
RS
R/W
E
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
LED+
LED-
LCD

Il codice

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```