

18 – Il display a cristalli liquidi 1602 – the 1602 display (some notes at section end)



Il display 1602 e' una versatile interfaccia di output sulla quale possono essere visualizzati messaggi di due righe da 16 caratteri. Il modulo e' facilmente gestibile attraverso il controller Hitachi HD44780, emulato da Arduino tramite uno specifico software presente nelle sue librerie. Le funzionalita' dei 16 pin del display sono illustrate nella seguente tabella:

1	VSS - collegamento a terra
2	VDD - alimentazione 5 volt
3	V0 - regolazione del contrasto
4	RS – register select: High=dati, Low=istruzioni
5	R/W - High=Read; Low=Write
6	Abilita (High) o disabilita (Low) l'ingresso di un byte
7	D0 data bus - bit 0
8	D1 data bus - bit 1

9	D2 data bus - bit 2
10	D3 data bus - bit 3
11	D4 data bus - bit 4
12	D5 data bus - bit 5
13	D6 data bus - bit 6
14	D7 data bus - bit 7
15	BL-A retroilluminazione (5 volt)
16	BL-K retroilluminazione (gnd)

In linea di massima e' possibile dire che la configurazione dei pin da 7 a 14 (che possono essere HIGH oppure LOW e cioe' 0 oppure 1) rappresenta il byte che si intende visualizzare. Il display riceve, memorizza in un registro e visualizza il byte seguendo le istruzioni che gli vengono fornite attraverso i pin 4, 5 e 6. Piu' in dettaglio:

- I pin da 1 a 3 sono utilizzati per alimentare il display e per regolare (mediante un potenziometro) il contrasto dei caratteri
- Il pin 4 (register select) e' un selettore che, se dichiarato High predispone il registro del display a ricevere un byte mentre se dichiarato Low tiene il registro in attesa del prossimo evento.
- Il pin 5 stabilisce la modalita' di utilizzo del registro. Se e' LOW il registro e' in ricezione (write) mentre se e' HIGH il registro rende disponibile l'immagine del byte appena memorizzato. Se non si ha necessita' di interrogare il byte presente nel registro conviene mantenere questo pin costantemente LOW e quindi collegato a terra.
- Il pin 6 abilita o disabilita l'ingresso e la visualizzazione del byte
- I pin da 7 a 14 rappresentano, come detto, la configurazione del byte da visualizzare
- I pin 15 e 16 sono utilizzati per illuminare il fondo del display. Devono essere collegati all'alimentazione da 5 volt ed alla terra o, eventualmente, ad un interruttore. Possono anche ovviamente essere collegate ad una porta di Arduino e pilotate da programma (es. da una routine tipo luce crepuscolare)

Il display puo' lavorare in modalita' 8 bit oppure in modalita' 4 bit; se lavora in modalita' 8 bit puo' visualizzare tutto il set di caratteri ascii (maiuscole, minuscole e caratteri speciali), ma richiede l'utilizzo di 10 porte mentre se lavora in modalita' 4 bit utilizza solo 6 porte (4 per i dati, una per il selettore ed una per abilitare/disabilitare la visualizzazione). Per mostrare un testo formato da numeri, caratteri maiuscoli o minuscoli e punteggiatura e' sufficiente utilizzare la modalita' a 4 bit.

In questo esercizio ci limiteremo a visualizzare un breve testo e dei numeri, per cui sara' sufficiente utilizzare solo 4 porte per la trasmissione dei dati e quindi solo 6 porte di Arduino.

Per provare tutte le funzionalita' "fisiche" del display aggiungeremo anche un potenziometro collegato al pin 3 in modo da poter regolare il contrasto, ed un circuito per l'alimentazione della luminosita' dello sfondo, collegato ai pin 15 e 16 e pilotato da un normale pulsante on/off.

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator



Display 1602 is a versatile output interface on which can be displayed messages on two lines by 16 characters each. The module is easily managed through the Hitachi HD44780 controller, emulated by Arduino through a specific software present in its libraries.

The 16-pin functionality is illustrated in the following table

1	VSS – connect to ground	9	D2 data bus - bit 2
2	VDD – to 5 volts cc	10	D3 data bus - bit 3
3	V0 - Contrast adjustment	11	D4 data bus - bit 4
4	RS – register select: High=data, Low=rules	12	D5 data bus - bit 5
5	R/W - High=Read; Low=Write	13	D6 data bus - bit 6
6	Enable (High) or disable (Low) a byte input	14	D7 data bus - bit 7
7	D0 data bus - bit 0	15	BL-A Backlighting (5 volt)
8	D1 data bus - bit 1	16	BL-K Backlighting (gnd)

In general we can say that the configuration of pins 7 to 14 (which can be HIGH or LOW, ie 0 or 1) represents the byte you want to display. The display receives, stores in a log and displays byte, following rules provided through the pins 4, 5 and 6. In detail:

- Pins from 1 to 3 are used to power the display and to adjust (by a potentiometer) the characters contrast
- Pin 4 (register select) is a selector that, if declared HIGH, sets the display register to receive a byte; while LOW, sets register waiting for the next event.
- Pin 5 establishes how to use the register. If 'LOW' the register is in write and if HIGH makes available the newly stored byte image. If you do not need to interrogate the byte present in the register, you should keep this pin constantly LOW and then connect it to ground.
- Pin 6 enables or disables the input and display of the byte
- Pins 7 to 14 represent, as mentioned, the configuration of the byte to be displayed
- Pin 15 and 16 are used for backlight. They must be connected to the a 5 volt power supply and to ground and, if necessary, to a switch. They can of course also be connected to an Arduino pin and handled by a program (eg by a twilight routine)

Display can work in 8 bit mode or in 4 bit mode; If it works in 8-bit mode, it can display the entire ASCII set (uppercase, lower case, and special characters), but requires 10 Arduino pins, while if it works in 4-bit mode it uses only 6 pins (4 for data , one for the selector, and one to enable and disable the display). To show a text consisting of numbers, uppercase or lower case, and punctuation, just use the 4-bit modality.

In this project we will limit ourselves to displaying a short text and numbers, so we will use only 4 ports for data transmission and only 6 Arduino pins.

To test all the "physical" functions, we will also add a potentiometer connected to pin 3 to adjust contrast and a background power supply circuit connected to pins 15 and 16 and driven by a on / off button

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

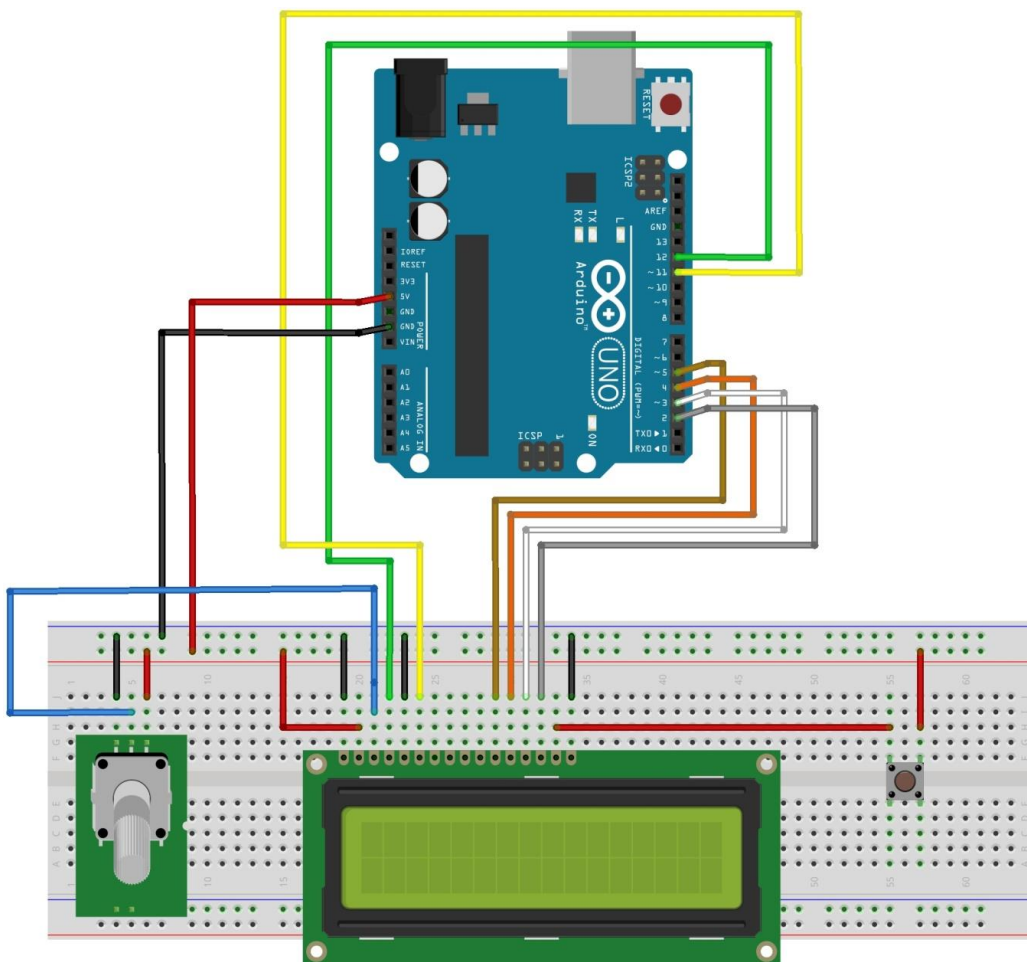
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

Materiali

- Un display lcd modello 1602
- Un potenziometro da 10 k
- Un interruttore
- Una breadboard
- Un po' di cavi

Schema



Programma

Download del programma tramite [Dropbox](#)
Program download via [Dropbox](#)

/ Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il trasferimento nell'IDE, premendo CTRL+T. In questo esercizio si utilizza il display a cristalli liquidi 1602. Il programma utilizza le funzioni rese disponibili dalla libreria "LiquidCrystal" che emula il driver Hitachi HD44780*

Il circuito prevede le seguenti connessioni:

- * LCD VSS (pin 1) connesso a terra (gnd)*
- * LCD VDD (pin 2) connesso all'alimentazione di 5 volt*
- * LCD V0 (pin 3) connesso al polo intermedio di un potenziometro da 10 kohm i cui estremi sono collegati, rispettivamente, a terra ed all'alimentazione da 5 volt*
- * LCD RS (pin 4) connesso alla porta 12 di Arduino*
- * LCD R/W (pin 5) connesso a terra (gnd)*
- * LCD Enable (pin 6) connesso alla porta 11*
- * LCD D4 (pin 11) connesso alla porta 5*
- * LCD D5 (pin 12) connesso alla porta 4*
- * LCD D6 (pin 13) connesso alla porta 3*
- * LCD D7 (pin 14) connesso alla porta 2*
- * LCD BL-A (pin 15) connesso all'alimentazione da 5 volt (o ad un interruttore collegato all'alimentazione da 5 volt)*
- * LCD BL-K (pin 16) connesso a terra (gnd)*

Sul display verra' visualizzata la parola "Buongiorno!", seguita da un contatore che si incrementa di 1 ad ogni secondo

** Warning: cut&paste from PDF to IDE loses formatting. To restore it press CTRL + T. This project uses a liquid crystal display 1602. The program uses the functions available on the "LiquidCrystal" library, that emulates the Hitachi HD44780 driver*

Here the connections:

- * LCD VSS (pin 1) to Arduino ground*
- * LCD VDD (pin 2) to Arduino 5 volt pin*
- * LCD V0 (pin 3) Connected to the intermediate pole of a 10k ohm potentiometer whose ends are respectively connected to ground and to the Arduino 5 volt pin*
- * LCD RS (pin 4) to Arduino pin 12*
- * LCD R/W (pin 5) to Arduino ground*
- * LCD Enable (pin 6) to Arduino pin 11*
- * LCD D4 (pin 11) to Arduino pin 5*
- * LCD D5 (pin 12) to Arduino pin 4*
- * LCD D6 (pin 13) to Arduino pin 3*
- * LCD D7 (pin 14) to Arduino pin 2*
- * LCD BL-A (pin 15) to Arduino 5 volt pin or to an ON/OFF button, connected to Arduino 5 volt pin*
- * LCD BL-K (pin 16) to ground*

The word "buongiorno!" Will be displayed on display, followed by a counter that is incremented by 1 every second.

**/*

`#include <LiquidCrystal.h> /* dichiarazione di utilizzo della libreria "LiquidCrystal"*/`

`LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // inizializza la libreria con i numeri delle porte usate`

`void setup()`

```
{
  lcd.begin(16, 2); // definisce il numero di caratteri per riga ed il numero di righe da gestire
  lcd.print("Buongiorno!"); //visualizza sul display la parola "Buongiorno!"
}
```

`void loop()`

```
{
  lcd.setCursor(5, 1);        // Posiziona il cursore sulla colonna 5 della riga 1
  // attenzione: la riga 1 e' di fatto la seconda riga del display poiche' la prima e' la riga 0,
  // quella sulla quale abbiamo scritto "buongiorno!"
  lcd.print(millis()/1000); // visualizza i secondi trascorsi dall'avvio o dal reset
}
```