



Paolo Rognoni (Paolino)

LO HAI MAI REALIZZATO CON UN PIC? - I PICMICRO E I DISPLAY LCD ALFANUMERICI (PARTE 1)

28 January 2012

Dopo aver scritto diversi articoli, ritengo che sia il caso di iniziare a proporre qualcosa di un po' più "interessante" per la rassegna "**Lo hai mai realizzato con un PIC?**". Non che io creda che quanto pubblicato fino ad ora non lo sia stati, ma forse è il caso di proporre qualcosa di più accattivante. Nell'ottica di far precedere contributi propedeutici ad articoli più applicativi, voglio raccontare qualcosa sui **display LCD**. Ma come, un altro articolo sui display LCD? Sì, certo, cercando di porre l'accento su alcuni aspetti che secondo me sono rilevanti, tra cui l'utilizzo di un compilatore open source e la portabilità del codice. In questa raccolta si partirà con i display alfanumerici per poi, tempo permettendo, arrivare a quelli grafici.

Fino a qualche anno fa, un po' per pigrizia, un po' per comodità e un po' per pragmatismo, quando mi sono trovato a dover utilizzare display LCD mi sono sempre cercato delle librerie già pronte. Ma una volta accadde che il compilatore non metteva a disposizione proprio nulla per l'interfacciamento di periferiche che non fossero strettamente legate al micro, mentre il progetto che stavo mettendo su carta aveva tassativamente bisogno di un LCD 16x2. Mi sono quindi trovato nella necessità di dover sviluppare un semplice driver e ho cercato di scriverlo in modo che un giorno potessi riutilizzarlo. Quella volta usavo un microcontrollore di Fujitsu, non PICMicro.

I display LCD ed il controller HD44780

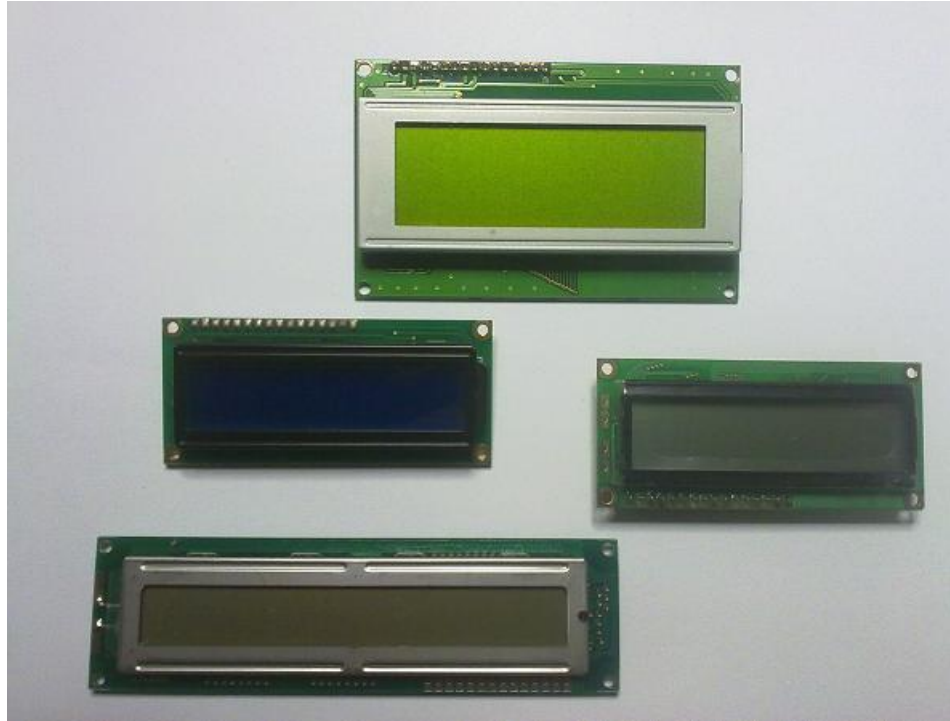
Quello che si vede in fotografia è un classico display a 4 righe, con 20 caratteri per riga; in gergo un 20x4.



Un LCD 20x4

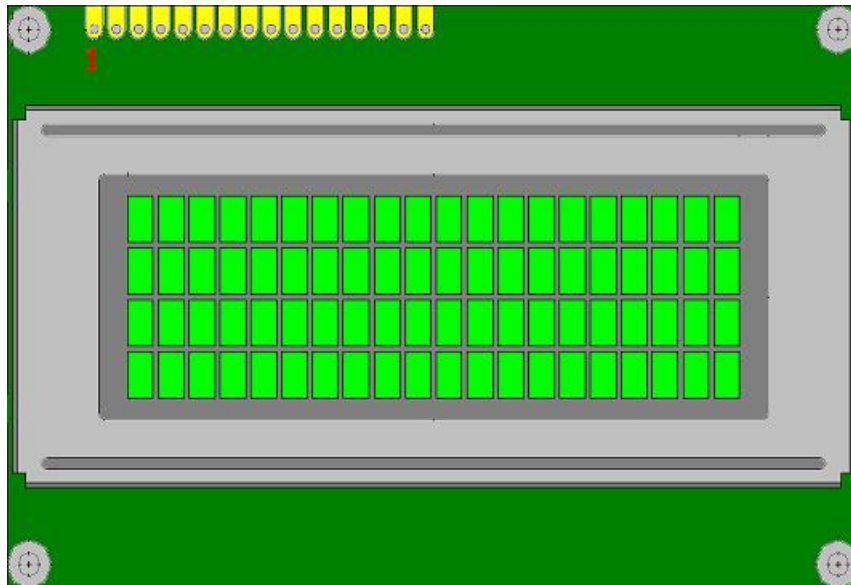
Tipicamente, display di questo tipo sono costruiti da Case differenti ma tutti hanno in comune un aspetto: il controller utilizzato. Una ventina di anni fa, quando questi display iniziavano a fare capolino nelle apparecchiature di uso quotidiano, tutti i Costruttori di LCD utilizzavano il medesimo controller, il blasonato HD44780 di Hitachi. Oggi, sono disponibili "cloni" di quel controller, prodotti da diversi Costruttori. Il vantaggio per i progettisti (tanto hardware quanto firmware) è che la compatibilità con **HD44780** permette la sostituzione di esemplari di LCD con altri più economici, senza doversi preoccupare più di tanto (su questo aspetto si potrebbe aprire un bel dibattito, ma non è oggetto di questo articolo). Fatto sta, che oggi i controller sono compatibili con il capostipite di Hitachi.

Grazie a questo aspetto, il firmware che si scrive per interfacciare un particolare LCD generalmente è portabile quando si cambia modello di LCD; il controller, infatti, dialoga con il microcontrollore per mezzo di comandi impartiti secondo criteri stabiliti dalle tabelle che sono presenti nei datasheet. Ora, quando si acquista un componente o un dispositivo, è doveroso disporre anche del datasheet. Per i display LCD la necessità si fa oltremodo necessaria, in quanto la disposizione dei pin del display potrebbe non essere sempre la stessa, da display a display. In fotografia riporto alcuni display che montano tutti il medesimo controller ma che hanno pinout differente.



Quattro LCD. A partire dall'alto: LCD20x4 verde retroilluminato, LCD16x2 blu retroilluminato, LCD16x2 blu non retroilluminato, LCD16x1 blu non retroilluminato

Il display LCD che ho considerato per questo mio articolo è schematizzato qui in figura, dove si nota l'indicazione del pin 1



LCD 20x4 schematizzato

Il significato dei pin è quello illustrato nella seguente tabella:

Numero pin	Simbolo	Descrizione
1	Vss	Alimentazione: 0V
2	Vdd	Alimentazione: 5V
3	Vo	Regolazione contrasto
4	RS	Dati o comandi
5	R/W	Lettura o scrittura
6	E	Enable (chip Enable)
7	DB0	Data Bus, bit 0
8	DB1	Data Bus, bit 1
9	DB2	Data Bus, bit 2
10	DB3	Data Bus, bit 3
11	DB4	Data Bus, bit 4
12	DB5	Data Bus, bit 5
13	DB6	Data Bus, bit 6
14	DB7	Data Bus, bit 7
15	A	Retroilluminazione, anodo
16	K	Retroilluminazione, catodo

I sedici pin possono essere raggruppati secondo la loro funzionalità:

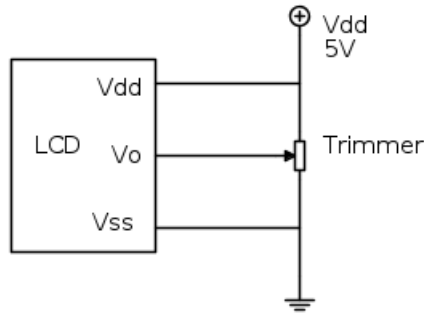
- **alimentazione:** 1, 2;
- **regolazione contrasto:** 3;
- **databus:** 7, 8, 9, 10, 11, 12, 13, 14
- **pin di controllo:** 4, 5, 6;
- **retroilluminazione:** 15, 16

Alimentazione

Questi display tipicamente si alimentano a 5V; l'alimentazione principale non attiva la retroilluminazione che viene comandata a parte (vedi paragrafo Retroilluminazione). La corrente assorbita dalla logica dipende molto da quanto sia grande (in termini di caratteri) il display e da quanta potenza sia richiesta per accendere i pixel.

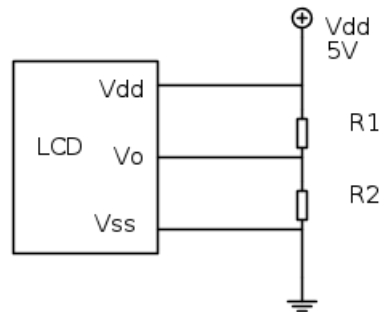
Regolazione del contrasto

I caratteri visualizzati possono essere poco o troppo contrastati con lo sfondo. Per modificare la regolazione del contrasto si utilizza generalmente un trimmer collegato come in figura. Ci sono sistemi alternativi all'utilizzo del trimmer, come ad esempio l'impiego di un DAC. Fatto è che il contrasto è funzione della tensione applicata al pin Vo



Regolazione contrasto con trimmer

In alternativa è possibile *bloccare* il contrasto, utilizzando due resistori di valore noto, evitando così l'utilizzo di un trimmer. Questa operazione può essere critica per diversi aspetti: con la temperatura ambiente il contrasto può variare anche sensibilmente e potrebbe essere richiesto un "ritocco"; inoltre la temperatura agisce anche sui valori dei resistori andando a modificare il valore del partitore. Per produzioni di serie, spesso questa soluzione è quella preferita per i notevoli vantaggi che si hanno in termine di costo.



Regolazione contrasto con partitore

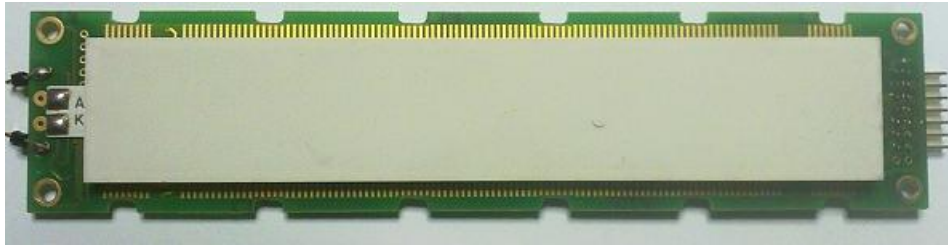
Per adottare la soluzione con due resistori, si può facilmente partire dalla soluzione con trimmer, regolare la posizione del cursore affinché si abbia il contrasto desiderato, e misurare la tensione sul pin V_0 riferita a V_{SS} . A partire dalla formula del partitore di tensione:

$$V_0 = R_2 \frac{V_{dd}}{R_1 + R_2}$$

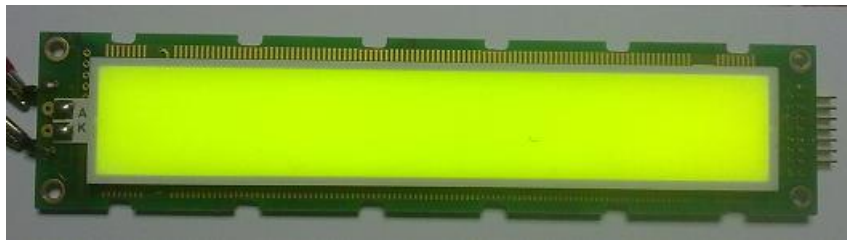
impostando ad esempio il valore di R_1 , si ricava il valore di R_2 . Si sceglierà poi il valore commerciale più vicino.

Retroilluminazione

Questa è una caratteristica non sempre disponibile; non tutti i display ne sono dotati. Qualora ci sia, la retroilluminazione è ottenuta con LED; la luce generata viene fatta passare attraverso un diffusore che omogenizza il più possibile la luce, "rompendo" i raggi prodotti dai LED. Nel seguito, si illustra un modulo di retroilluminazione spento ed acceso, con particolare attenzione all'assorbimento.



Modulo di retroilluminazione spento



Modulo di retroilluminazione acceso



Assorbimento della retroilluminazione

Trattasi di modulo di retroilluminazione di un LCD 20x2 "maxi", pertanto non c'è da stupirsi se l'assorbimento supera i 500mA.

Anche in questo caso la lettura del datasheet è a dir poco fondamentale: l'adozione di una resistenza di limitazione della corrente e/o di un transistor comandato in saturazione potrebbe essere necessario, qualora gli assorbimenti fossero notevoli.

Modalità di connessione: il databus e i pin di controllo

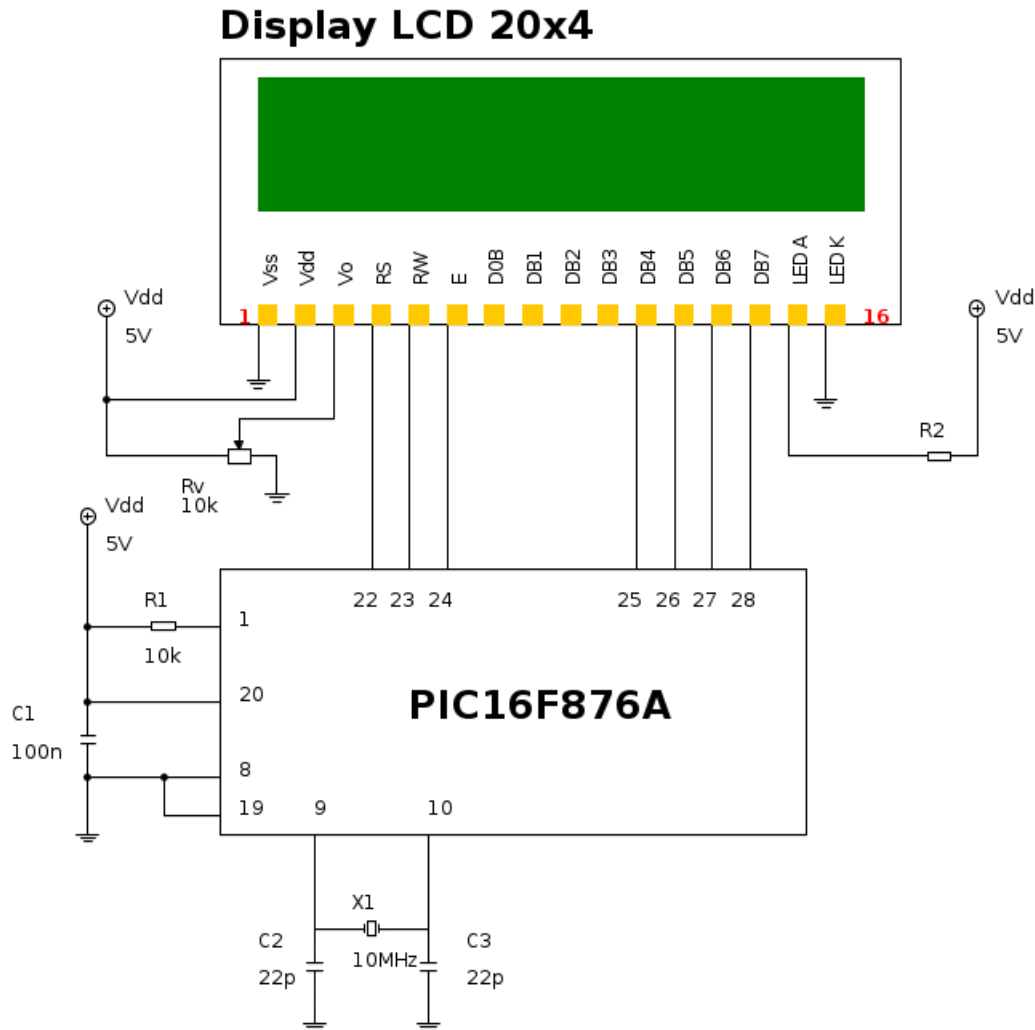
Il controller HD44780 permette due modalità di connessione al microcontrollore:

- con databus a 8 bit;
- con databus a 4 bit.

I due casi si distinguono sia per il numero di pin impiegati sia per le diverse impostazioni necessarie al firmware. Quello che rimane però in comune alle due modalità è il numero di pin di controllo, che restano sempre tre. In particolare:

- E: chip **E**nable
- R/W: **R**ead or **W**rite (lettura R o scrittura W)
- RS: **R**egister **S**elect (comandi o dati)

Non c'è dubbio sul fatto che la modalità con databus a 4 bit permette di risparmiare sul numero dei pin del microcontrollore, anche se il firmware (come vedremo successivamente) sarà sensibilmente più complesso, non complicato. Per questa ragione, nel seguito si andrà ad utilizzare un display connesso ad un PIC utilizzando solo 4 bit del databus; lo schema è quello riportato in figura.



Schema elettrico

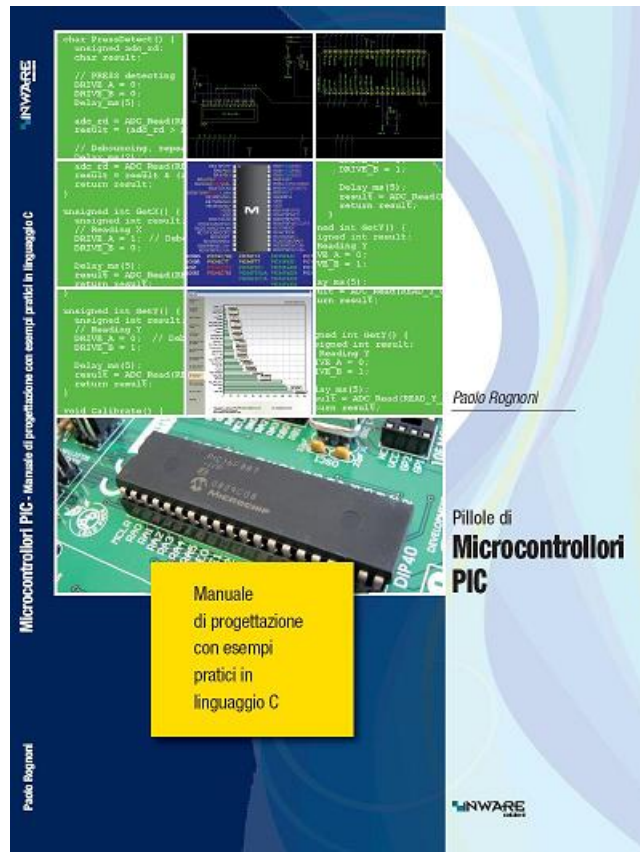
Per l'esempio, ho utilizzato un PIC16F876A, con quarzo da 10MHz; la scelta del PIC è puramente casuale (quello che avevo in casa, pronto all'uso). L'adattamento ad altri PIC rappresenta un esercizio che anche un neofita può affrontare, ovviamente disponendo di datasheet del microcontrollore.

Nella seconda parte...

Dopo questa breve introduzione, nel prossimo articolo verrà illustrato come pilotare il display con la modalità a 4 bit, la tabella dei comandi. Non resta che aspettare solo un pochino...

Bibliografia

- **Datasheet PIC16F876A:** <http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>
- **Il controller HD44780:** http://en.wikipedia.org/wiki/Hitachi_HD44780_LCD_controller
- **Hitachi Semiconductor:** <http://www.hitachi.com/product/business/semicon/index.html>
- **Ambiente di sviluppo MPLAB:** <http://www.microchip.com/mplab>
- **Compilatore SDCC:** <http://sdcc.sourceforge.net>
- **GPUTILS:** <http://gputils.sourceforge.net>
- **Sito web PicExperience**
- **Collana "LO HAI MAI REALIZZATO CON UN PIC?":**
 - [Il contamarce](#)
 - [Una sorpresa musicale per Babbo Natale](#)
 - [Una tecnica antirimbalzo](#)
 - [Il dado elettronico](#)
 - [Un approccio ai timer dei PICMicro](#)
 - [I PIC e i segnali analogici: la conversione A/D](#)
 - [Generare segnali PWM con i PICMicro](#)
 - [La lampada SIBILLA!](#)
 - [Una tecnica di misura della tensione di batteria](#)
 - [Il PIC tiene la data e l'ora](#)
 - [Una prima occhiata a SDCC](#)
 - [PIC Watch un semplice OROLOGIO SVEGLIA](#)
- **[Pillole di microcontrollori PIC:](#)**



Pillole di Microcontrollori PIC

Estratto da ["http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:lo-hai-mai-realizzato-con-un-pic-i-picmicro-e-i-display-lcd-alfanumerici-parte-1"](http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:lo-hai-mai-realizzato-con-un-pic-i-picmicro-e-i-display-lcd-alfanumerici-parte-1)