



Set didattici per sviluppare  
con **Arduino** e **Raspberry Pi**

[home](#) | [area personale](#) | [schemi](#) | [tutorial](#) | [robotica](#) | [pic micro](#) | [recensioni](#) | [forum](#) | [chat irc](#) | [faq](#) | [contatti](#) | [store](#)

logout

micciotta62 - ultimo accesso il 05/05/2015, 09:26

non hai nuovi messaggi

cerca

## LED DISPLAY TUTORIAL

Teoria e pratica dei display e matrici (parte prima)

tipo:

livello:

# LED DISPLAY TUTORIAL

Teoria e pratica sull'utilizzo dei display e delle matrici

## PARTE PRIMA

### Premessa:

Questo tutorial è frutto di esperienze e prove sull'uso dei più comuni display a sette segmenti o display a matrice di punti (8x8 o 7x5). E' rivolto principalmente a tutti coloro che iniziano o hanno voglia di sperimentare per apprendere e accrescere il proprio bagaglio tecnologico. In questa prima parte analizzeremo le componenti HW di questi dispositivi mentre nella seconda parte verranno analizzati alcuni software specifici in ambiente "Arduino".

### Descrizione:

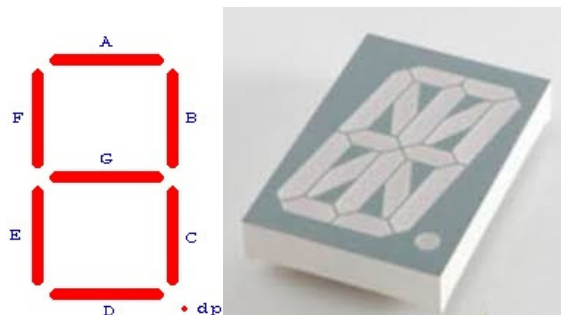
L' utilizzo dei display a 7 segmenti è abbastanza semplice e vengono utilizzati in moltissimi strumenti per la visualizzazione numerica di misure varie. Oggi possiamo vederne di tutte le dimensioni e colori (verde, rosso, giallo) usati anche per orologi digitali di grandi dimensioni. Non dimentichiamo poi l'utilizzo su strumenti di vario tipo: insomma sono dappertutto e in forme diverse. Non confondiamo però questi display con i pannelli a led o a cristalli liquidi con simboli diversi integrati. Prendiamo ad esempio un tachimetro digitale che mostra i valori come se fosse uno strumento analogico con tanto di lancette ma non possiamo definirlo a 7 segmenti.

Ma restiamo sul semplice e analizziamo i modelli più comuni alla portata di qualsiasi hobbista, sperimentatore o, come il sottoscritto, curioso di approfondire.

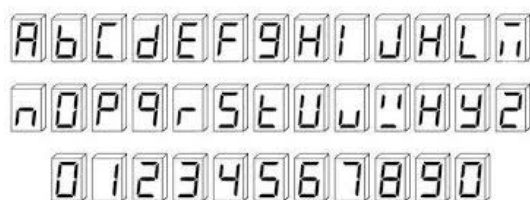
## DISPLAY A 7 SEGMENTI SINGOLI:

Sono i più comuni e di dimensioni contenute anche se in commercio esistono i cosiddetti display "giganti" di grandi dimensioni per pannelli esterni.

Sono detti a 7 segmenti in quanto utilizzano dei piccoli segmenti indipendenti a formare un "8" leggermente inclinato. La forma di questi segmenti può essere diversa allo scopo di formare un numero maggiormente e gradevolmente visibile. Esiste però un "ottavo" segmento che è in realtà un "punto decimale" che di solito si trova sul lato inferiore destro del display. Ogni segmento viene, per convenzione, definito con una lettera dell'alfabeto da "A" a "G" mentre il punto decimale viene definito come "DP" secondo questo schema.



Le due immagini mostrano un semplice display a 7 segmenti e le relative definizioni dei segmenti e un display multi-segmenti (se ne possono contare 16) specifico per comporre lettere alfabetiche più precise o altri simboli.



Come evidenziato dalle figure soprastanti con i sette segmenti si possono raffigurare tutti i numeri decimali ed un set alfabetico quasi completo anche se qualche lettera ha bisogno di una interpretazione (vedere la "v", "w", "m"). Quindi accendendo una serie di segmenti si ottiene la cifra o la lettera voluta. Ad esempio per comporre il numero "0" occorre accendere tutti i segmenti eccetto il segmento "G". Ciò è da tenere presente quando parleremo del software di gestione.

## DISPLAY MULTIPLI:

Non sono differenti dai display singoli ma hanno la caratteristica di essere inglobati in un unico contenitore (due, tre, quattro o più) per formare un display a più cifre. Il pilotaggio è identico ai display singoli ma con una variante: tutti i segmenti sono connessi tra loro in parallelo. A prima vista ci si potrebbe chiedere: ma se accendo il segmento "A" si accendono tutto gli altri? Risposta: **NO** perché ogni singolo display è pilotabile singolarmente. La risposta è semplicistica perché in realtà il pilotaggio è un po' più complicato.

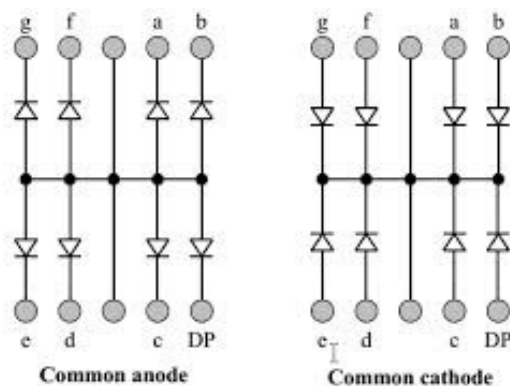


## LE CONNESSIONI ELETTRICHE:

Abbiamo visto che ogni display presenta 8 connessioni relative ai segmenti ma non abbiamo ancora parlato di come accendere questi segmenti. Esiste allora uno specifico **"PIN"** che possiamo definire *"pin di pilotaggio"* .

Sicuramente saprete che i display vengono definiti ad **"ANODO COMUNE"** oppure a **"CATODO COMUNE"**. Questa differenza determina la connessione elettrica per l'accensione dei segmenti. Si definisce un display a **"CATODO COMUNE"** quando per accendere i segmenti occorre portare a *"massa"* il cosiddetto *"pin di pilotaggio"* e al polo positivo (con una resistenza da 220/330 ohm) un qualsiasi segmento. Viceversa se dobbiamo pilotare un display a **"ANODO COMUNE"**.

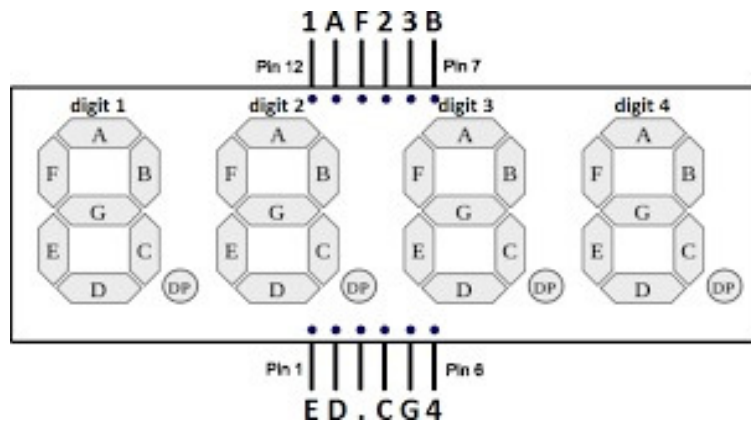
La figura sottostante dimostra le connessioni:



Questa differenziazione è molto importante in quanto determina poi la scelta dei componenti per lo sviluppo dello schema elettrico.

## ESEMPIO PRATICO:

Prendiamo un multi\_display in quanto ci può facilitare la comprensione di come connettere e utilizzare questi display: vediamo la figura di uno di questi:



Come ho detto in precedenza nei display multipli tutti i segmenti sono connessi tra loro e ogni display ha un suo specifico pin di pilotaggio. (Definito come "A" se a anodo comune o "K" se a catodo comune). Nella figura questo display ha 12 piedini: 8 per i segmenti e 4 per il pilotaggio numerati da 1 a 4. Non è definito se sono a anodo comune o catodo comune ma per scoprirlo è sufficiente effettuare due connessioni elettriche secondo quanto detto in precedenza.

Se invece di avere un multi\_display abbiamo singoli display il concetto non cambia: connettiamo fisicamente in parallelo tutti i segmenti (8) e isoliamo il pin di pilotaggio.

Ma sia per quanto riguarda i display multipli che quelli singoli connessi tra loro, esiste un problema semplice ma che a prima vista (almeno per i principianti) non si evidenzia.

Poniamo di voler accendere il numero "0" sul primo display. Nessun problema: connettiamo l'alimentazione rispettando le polarità del display (segmenti e pin di pilotaggio) e vedremo accendersi il numero. Perfetto prova riuscita !.

Ma se vogliamo accendere anche il secondo display "**contemporaneamente al primo**" con il numero "1" ?. Dovremmo fare la stessa cosa che abbiamo fatto per il numero "0": invece no! non funziona. Nel migliore dei casi vedremo accendersi solo il primo o il secondo display ma "**non contemporaneamente**" oppure potrebbe capitare di vedere lo stesso numero su entrambi i display .

Il perché è presto detto: Per far sì che si possano visualizzare "insieme" uno o più numeri sui display occorre adottare la tecnica definita "**multiplexing**". Questa tecnica di solito viene effettuata da software e consiste nel creare una routine che permetta di accedere i display voluti con il numero voluto in "**continuazione**" con un livello di ritardo tra l'accensione e l'altra di ogni display tale da ingannare l'occhio umano creando la persistenza di immagine. Questo inganno è facilmente provabile: è sufficiente riprendere con una telecamera ad una velocità maggiore del normale, un display in funzione. Nel migliore dei casi vedrete un solo display acceso, nel peggiore uno sfarfallio continuo. E a volte si può vedere anche a occhio nudo.

## DALLA TEORIA ALLA PRATICA:

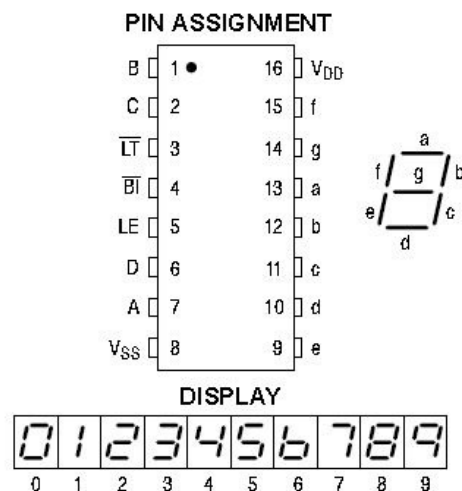
Facciamo un esempio tra i più classici: un orologio digitale a quattro cifre (ore e minuti) o a sei cifre (ore, minuti, secondi). Prendiamo 4(6) display a catodo comune (o anodo comune) e progettiamo un circuito adatto.

Per gestire questi display occorrono: 8 segnali per i segmenti A:G e DP e 4 (6) segnali per il pilotaggio degli stessi. Quindi in totale ci servono da 12 ai 14 fili di connessione ed altrettanti "pin" disponibili di un qualsiasi microcontroller. (salvo non fare un circuito specifico autonomo con tanto di clock per l'orologio le varie decodifiche ecc. ecc.). L'utilizzo di un microcontroller sicuramente facilita il progetto a patto che si sappia come programmarlo. Ma ritengo che questo non sia più, oggi, un problema anche per i principianti. Ma 14 o 16 fili sono tanti anche per i più potenti microcontroller: dobbiamo cercare di ridurre queste connessioni al minimo indispensabile e per ottenere questo risultato dobbiamo ricorrere alle cosiddette "decodifiche". Queste non sono altro che integrati che permettono dato un input di ottenere un output "decodificato". Vediamo come:

**Pilotaggio dei segmenti:** occorrono almeno 7 bit (quindi un byte) per settare nel modo corretto i segmenti ed ottenere il numero corretto. La tabella seguente mostra come devono essere attivati i vari segmenti relativamente ad ogni numero o lettera. Si presuppone l'utilizzo di display a "catodo comune".

Digit Shown	Illuminated Segment						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Questa tabella è leggermente riduttiva in quanto non va oltre il numero "9" mentre è possibile visualizzare anche le lettere da A:F semplicemente ponendo a "1" o a "0" il segmento relativo. Un'altra cosa da tener presente è il peso dei bit: di solito il bit relativo al segmento "A" è il bit "0" seguito dal bit "1" per il segmento "B" e così via di seguito. Per fare tutto questo abbiamo bisogno di una decodifica come questa in figura:



Trattasi del chip **4511** definito come "BCD\_TO\_SEVEN\_SEGMENT Latch/Decoder/Driver. Ovvero pilota decodificatore, driver, dal codice **bcd** al codice 7 segmenti. Vediamo la sua

“truth table”:

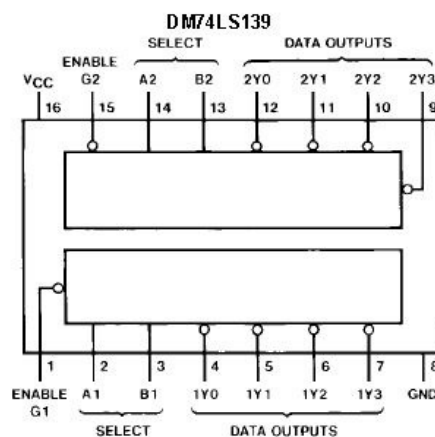
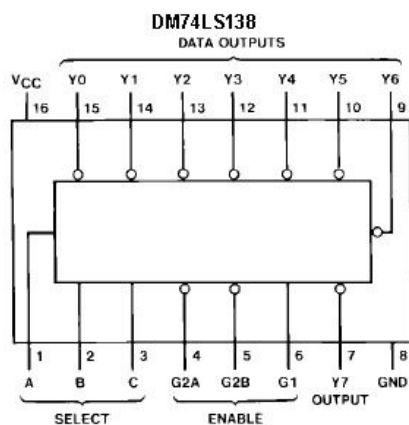
Inputs								Outputs						
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	Blank
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	1	1	0	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	0	1	1	0	0	0	0	0	0	0	Blank
0	1	1	1	1	0	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	0	0	0	0	0	0	0	0	Blank
1	1	1	X	X	X	X				*				*

X = Don't Care

\*Depends upon the BCD code previously applied when LE = 0

In sostanza fornendo agli ingressi “A:D” l’appropriato valore binario otteniamo alle uscite i 7 bit che pilotano i 7 segmenti. Tralascio per adesso gli altri pin in quanto l’unico che serve è il pin “LE” che deve essere a “0” logico per il corretto funzionamento del chip. L’uso di questa decodifica è limitato ai soli numeri mentre se avessimo bisogno anche delle lettere da “A” a “F” (per la tipica rappresentazione esadecimale di un byte) dovremmo utilizzare il chip 9368 che decodifica anche le suddette lettere.

**Pilotaggio dei display:** Con il sistema appena visto abbiamo risparmiato 3 fili su 7; adesso dobbiamo considerare il pilotaggio dei display. Abbiamo visto che per pilotare 4 display occorrono 4 fili (uno per ogni catodo/anodo). Anche in questo caso abbiamo la possibilità di risparmiare due fili su quattro utilizzando una decodifica specifica tipo **74LS138** o **74LS139** decoder/demultiplexer (anodo comune o catodo comune). Vediamo lo schema di questi due chip:



E la loro “truth table”:



## Function Tables

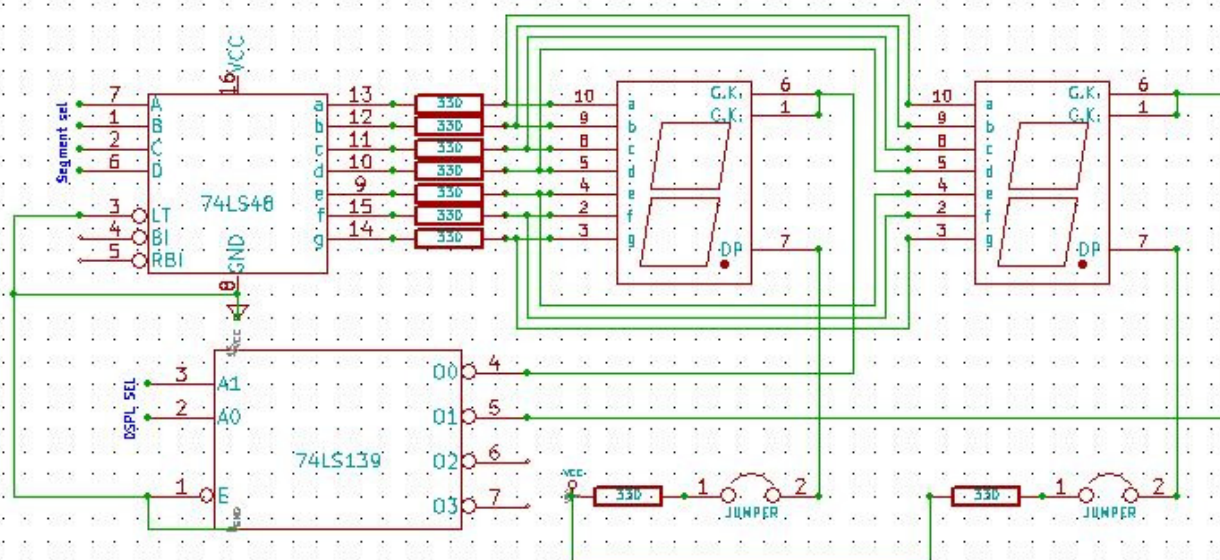
DM74LS138												DM74LS139							
Inputs		Outputs			Inputs		Outputs												
Enable		Select										Enable		Select			Outputs		
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	G	B	A	Y0	Y1	Y2	Y3
X	H	X	X	X	H	H	H	H	H	H	H	H	H	X	X	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H	L	L	L	L	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H	L	L	H	H	L	H	H
H	L	L	L	H	L	H	H	H	H	H	H	H	L	H	L	H	H	L	H
H	L	L	H	L	H	H	L	H	H	H	H	H	L	H	L	H	H	L	H
H	L	L	H	H	H	H	H	L	H	H	H	H	L	H	L	H	H	L	H
H	L	H	L	L	H	H	H	H	L	H	H	H	L	H	L	H	H	L	H
H	L	H	L	H	H	H	H	H	L	H	H	H	L	H	L	H	H	L	H
H	L	H	H	L	H	H	H	H	H	L	H	H	L	H	L	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	L	H	L	H	L	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L	L	L	L	H	H	H	H

H = HIGH Level  
L = LOW Level  
X = Don't Care

Note 1: G2 = G2A + G2B

In pratica ponendo opportunamente gli ingressi "A:B:C:" al loro valore binario otteniamo in uscita 7 bit diversi di cui uno soltanto a livello "H" o "L" in funzione del valore binario in ingresso. Con questo sistema con soli tre fili possiamo pilotare fino a 8 display diversi risparmiandone ben 4.

Lo schema seguente vuole essere solo un esempio pratico di come si possono connettere due display a 7 segmenti utilizzando componenti comuni. Lo schema presuppone l'utilizzo di display a "catodo comune". Deve essere ovvio che il relativo software di gestione deve prevedere la tecnica del "multiplexing". Resta inteso che ci sono altri sistemi per pilotare i display ricorrendo a tecniche software più complesse e che permettono di risparmiare ancora di più i "pin" da utilizzare.



Esempio di connessione per due display 7 seg

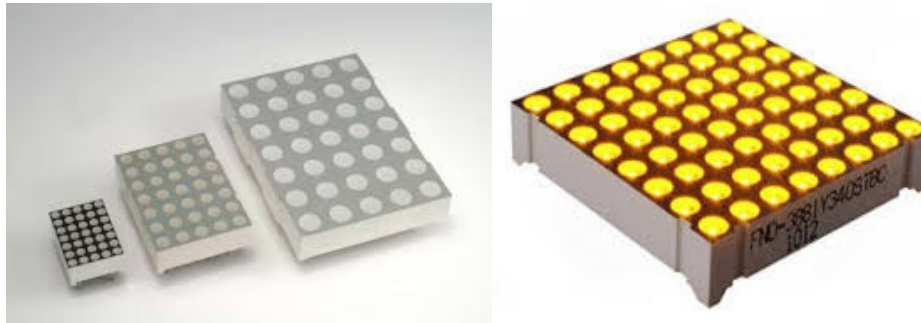
Un **74LS48** decodifica i 7 segmenti, un **74LS139** pilota i due display (con due pin liberi per pilotarne altri due). Un jumper opzionale serve a accendere il punto decimale non gestito.

Questo schema è sicuramente datato ma può servire come base per comprendere la logica di funzionamento. In realtà è possibile con soli 3 pin pilotare gli stessi display utilizzando

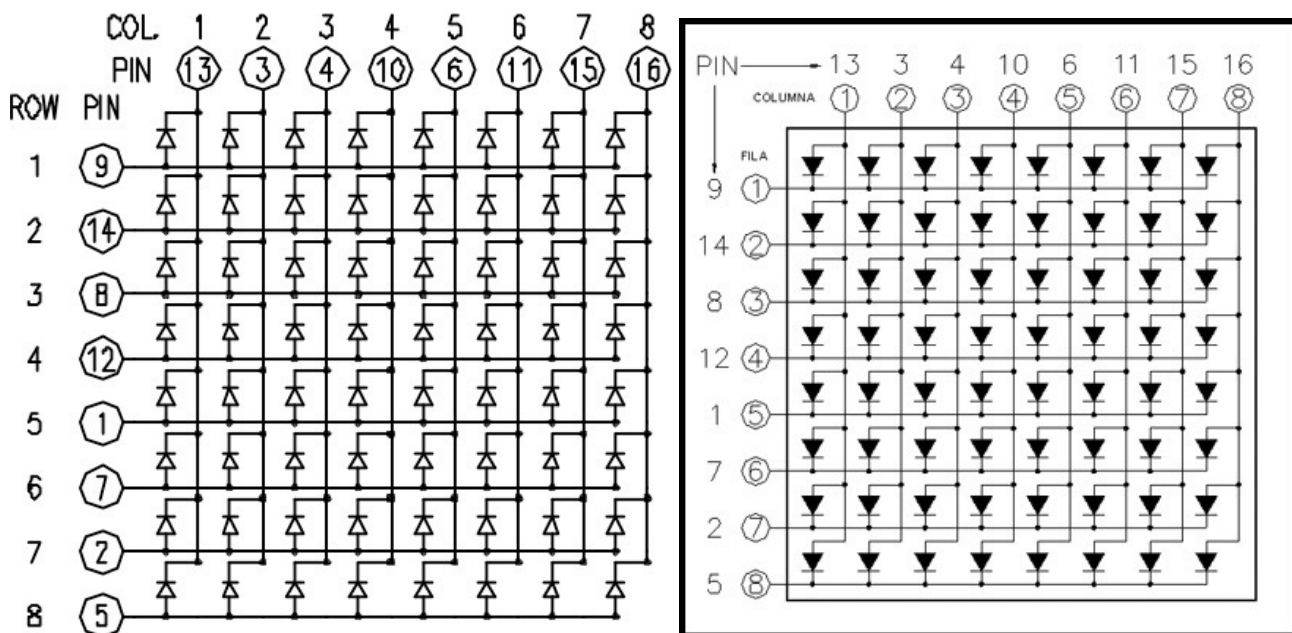
sempre la tecnica del multiplexing.

## I DISPLAY A MATRICE DI PUNTI:

Questi tipi di display, molto più performanti di quelli a 7 segmenti, sono sostanzialmente identici dal punto di vista elettrico: anche loro si dividono in due categorie (Anodo o catodo comune) ma invece di avere segmenti predefiniti presentano file di led disposti a quadrato: 8x8 o rettangolare 7x5 e con dimensioni diverse e con colori diversi e multidimensionali. Le figure sottostanti alcuni tipi.



La matrice a punti si identifica in righe e colonne: per le matrici più classiche 8 righe x 8 colonne nelle matrici 8x8 oppure 5 colonne per 7 righe nelle matrici 7x5; Questa suddivisione è molto importante per capire come pilotarle. Vediamo il loro schema elettrico (esempio per una matrice 8x8).



La figura di sinistra mostra che gli anodi dei led di ciascuna riga sono tutti connessi in parallelo analogamente ai catodi dei led di ciascuna colonna. Potremmo definire questa matrice ad **"ANODO COMUNE"**;

La figura di destra è l'esatto opposto: tutti i catodi dei led di una riga connessi in parallelo analogamente agli anodi dei led di ciascuna colonna. Potremmo definire questa matrice a **"CATODO COMUNE"**.



Ma, mentre per i display 7 segmenti la differenza tra Catodo o Anodo comune può essere ben determinata, nei display a matrice sembra un poco "ambigua". Ma è comunque necessaria in quanto poi determina il tipo di connessione elettrica da effettuare e gli eventuali integrati a supporto.

## PILOTAGGIO DELLE MATRICI:

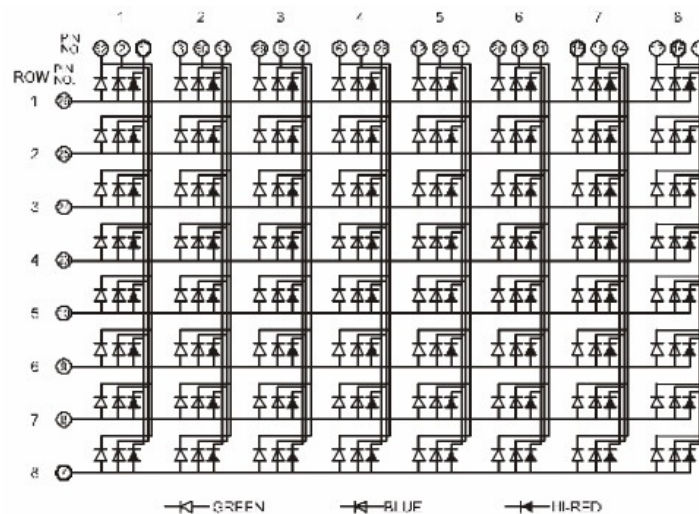
Il procedimento non si discosta molto da quello usato per i display a 7 segmenti. Con alcune varianti. Un singolo led si accende come se fosse un comunissimo led (colorato o meno): quindi sempre in dipendenza del tipo di matrice (Catodo/Anodo comune) per accendere un led occorrerà connettere la giusta polarità alla riga e alla colonna. Non dimenticandosi della resistenza di caduta (220/330 ohm) per evitare di bruciare i led.

Quindi una matrice 8x8 dispone di 64 led e questo permette di iscrivere in modo soddisfacente numeri, lettere, simboli, figure e creare vari effetti luminosi.

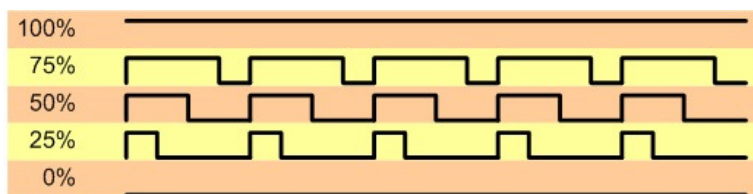
Nella gamma dei display a matrice sono presenti anche quelli a led RGB (Red,Green,Blue): ma in questo caso le cose si complicano in quanto ogni led presenta 3 pin rispettivamente per i colori RED, GREEN, BLUE oltre naturalmente il pin catodo/anodo.



La figura illustra un esempio. Ma la complessità circuitale è notevole:



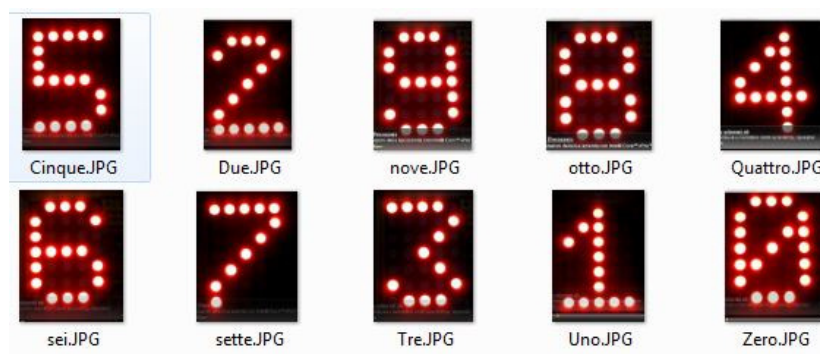
Questi display si prestano anche per essere utilizzati con la tecnica "PWM" acronimo di "Pulse Width Modulation". Questa tecnica permette di variare la luminosità dei singoli colori dei led tenendo presente anche la diversa luminosità intrinseca dei led. In pratica applicando al pin un treno di impulsi (HIGH\_LOW) in percentuale calibrata tra HIGH e LOW possiamo ottenere effetti luminosi diversi: dall'accensione allo spegnimento progressivo, il mix di colori ecc.. La tabella seguente dimostra come è possibile variare questi impulsi che, naturalmente, vengono gestiti via software. Attenzione però che in pratica non sarebbe possibile utilizzare il "PWM" per tutti i led: occorrerebbero 24 segnali diversi (3 per 8 led). Un numero troppo grande per essere gestito dai più comuni microcontroller.



Il diagramma di cui alla figura mostra la forma d'onda che può avere un segnale PWM: variandolo dal 100% allo 0% dipenderà la luminosità del led su cui è applicato.

Di seguito invece un piccolo esempio di utilizzo delle matrici 7x5: avendo solo 35 led disponibili la visione dei numeri o dei caratteri è più limitata. La costruzione dei numeri è una mia vecchia realizzazione di qualche anno fa attraverso un software su micro PIC.

Con una composizione di valori binari relativamente a ciascuna riga si ottiene l'effetto voluto. Ogni figura o numero da visualizzare sarà composto da una tabella (per le righe) di 7 elementi di 5 bit ciascuno: ogni bit "0" rappresenterà in led spento, ogni bit "1" un led acceso. Oppure con una tabella (per le colonne) di 5 elementi per 7 bit ciascuno. Sarà compito del software gestire opportunamente la visualizzazione.



Sicuramente pilotare un display a matrice di punti è più complesso rispetto ad un display a 7 segmenti. Se vogliamo fare un paragone è come avere 8(5) display connessi in parallelo tra loro. Con la differenza che la composizione di un numero o lettera o figura coinvolge l'intero insieme dei led che compongono la matrice. Quindi il software (o l'hardware) deve comporre una immagine dei 64 punti (nella matrice 8x8) che devono essere accesi o spenti per dare forma all'immagine. Anche in questo caso, come abbiamo già visto, è necessario utilizzare la tecnica del "multiplexing" .

## CONCLUSIONE DELLA PRIMA PARTE:

Quando descritto fino adesso non contempla sicuramente tutto il vasto mondo dei display ma vuole soltanto fornire un minimo livello di conoscenza: vagando in internet possiamo trovare migliaia di esempi, immagini, soluzioni, tutte diverse tra loro e tutte estremamente interessanti. Ma la base iniziale, non solo teorica e pratica, possono servire a capire e approfondire alcuni concetti base dell'intera materia.

La seconda parte di questo articolo sarà dedicata allo sviluppo software (e hardware) con riferimento all' ambiente "Arduino" con le diverse librerie a corredo.

Grazie per l'attenzione e alla seconda puntata.




Gvsoft

Aprile 2015

**il parere della community**

<b>esprimi il tuo voto</b>	<b>approvi questa pagina?</b>	<b>promo</b>
<p style="text-align: center; font-size: 24px; font-weight: bold;">9</p> <div style="display: flex; justify-content: center; gap: 5px;"> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> <div style="width: 15px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> </div> <p style="text-align: center; font-weight: bold;">numero di voti: 3</p> <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> <input type="button" value="vota"/> <input type="button" value="ok"/> </div> <p style="text-align: center; margin-top: 5px;"><a href="#">visualizza i voti</a></p> <hr style="border: 0.5px solid #ccc; margin: 10px 0;"/> <div style="display: flex; align-items: center; gap: 5px;"> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 5px; font-weight: bold;">pagina preferita</div> </div> <p style="font-size: 10px; margin-top: 5px;">6 utenti hanno questa pagina nei preferiti</p>	<div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <div style="text-align: center;"> <p style="font-weight: bold; color: green;">APPROVO</p> </div> <div style="text-align: center;"> <p style="font-weight: bold; color: red;">DISAPPROVO</p> </div> </div> <p style="font-size: 10px; text-align: center;">Il tuo giudizio <u>sarà reso pubblico</u> e potrai vedere cosa ne pensano gli altri utenti.</p>	

**lascia un commento o leggi i commenti presenti... [ 2 ]**

Autore	Messaggio	opzioni
<p>✉ <b>ad.noctis</b></p>  <p>★★★★★★★★</p> <p>10 10 10</p> <p>postato il: 13.05.2015, alle ore 23:09</p>	<p>Didatticamente parlando OTTIMO !</p>  <p>I miei studenti ne faranno tesoro.</p> <p>Grazie !</p> <hr/> <p>Marco</p>	
<p>✉ <b>gironico</b></p>  <p>★★★★★★★★</p> <p>10 10 10</p> <p>postato il: 14.05.2015, alle ore 07:49</p>	<p>mi piace, utile per me quello delle matrici, ancora non ho mai provato. Mi hai fatto venire voglia....A+9</p> <p>Piano bimbo fai piano. La Vigna è grande e il trattore è sempre quello che è!</p>	

[segui questo thread con grixFC](#), per questa funzione devi aver installato il [software grixFC](#)

[torna su](#)

[torna all'indice del forum](#) | [partecipa alla discussione](#)

### difficoltà



### costo



### informazioni

Questa pagina è stata creata da ✉ **gvsoft** ★★★★★★★★  
il 02/05/2015 ore 15:02  
ultima modifica del 04/05/2015 ore 13:42

la pagina è stata visitata 193 volte



Lo staff di [www.grix.it](#) non si assume responsabilità sul contenuto di questa pagina.

Se tu sei l'autore originale degli schemi o progetti proposti in questo articolo, e ritieni che siano stati violati i tuoi diritti, contatta lo staff di [www.grix.it](#) per segnalarlo.



[indietro](#) | [homepage](#) | [torna su](#)

copyright © 2004/2015 GRIX.IT - la community dell'elettronica amatoriale