

# Interfacciamento dei processori

## Terza Parte: Le periferiche di visualizzazione a 8 bit [1 di 3]

### PREMESSA

Nella **prima parte** di questa serie ci siamo occupati delle interfacce d'uscita per un dato a 8 bit, e abbiamo concluso che qualunque dispositivo programmabile (*microcontrollore* o *personal computer*) dispone *intrinsecamente* di almeno **almeno una** porta in grado di assicurare questo servizio.

Nella **seconda parte** abbiamo studiato le caratteristiche elettriche tipiche di una porta d'uscita, analizzando le specifiche di quella **TTL compatibili** (per altro le più diffuse) e proponendo una tecnica d'*analisi elettronica* facilmente generalizzabile anche per eventuali altre tecnologie. Abbiamo affrontato il concetto di **carico**, e proposto i primi due schemi d'interfaccia, uno con l'utilizzo dei **buffer di corrente non invertenti 74LS244** e uno con l'impiego delle **memorie esterne a 8 bit 74LS374** (*batterie di 8 flip-flop D-Type in parallelo*).

### PERIFERICHE A 8 BIT ADATTE ALLA GESTIONE DELL'INFORMAZIONE

Analizziamo ora le problematiche legate al progetto di **periferiche** adatte a tradurre visivamente le informazioni disponibili sulle uscite di un computer (*micro* o *PC*); le tratteremo con lo stile tipico di *chi non si ferma alle apparenze*, cercando di *aprire la mente* ai problemi meno evidenti.

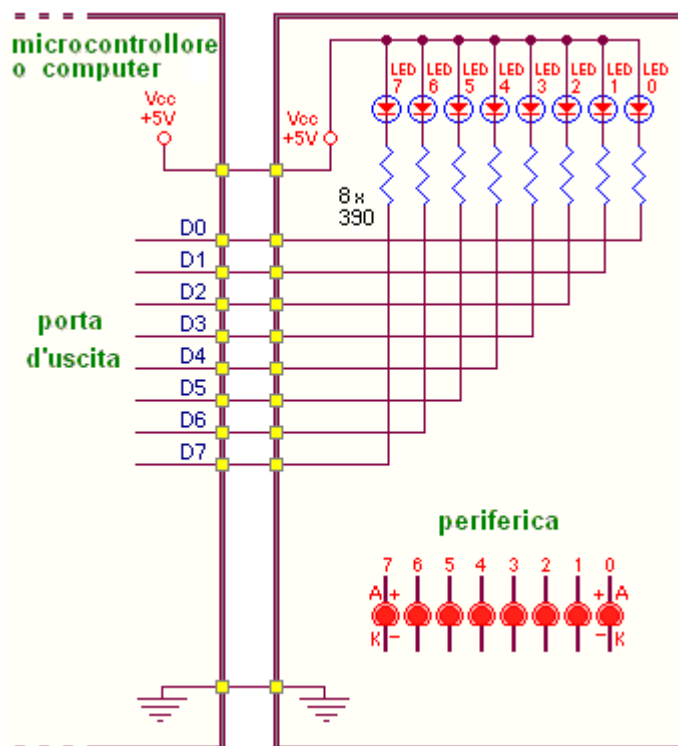
In tutti i progetti **faremo a meno** dello strato di **interfaccia** (costituito per esempio dal **74LS244** descritto in precedenza) rinunciando volutamente alla disponibilità sicura dei **15 mA erogabili** con uscita a "1" e dei **24 mA assorbibili** con uscita a "0", garantiti da questo importante componente; resta inteso che la loro eventuale presenza è tutt'altro che sgradita...

## Costruiamo la prima, indispensabile, periferica

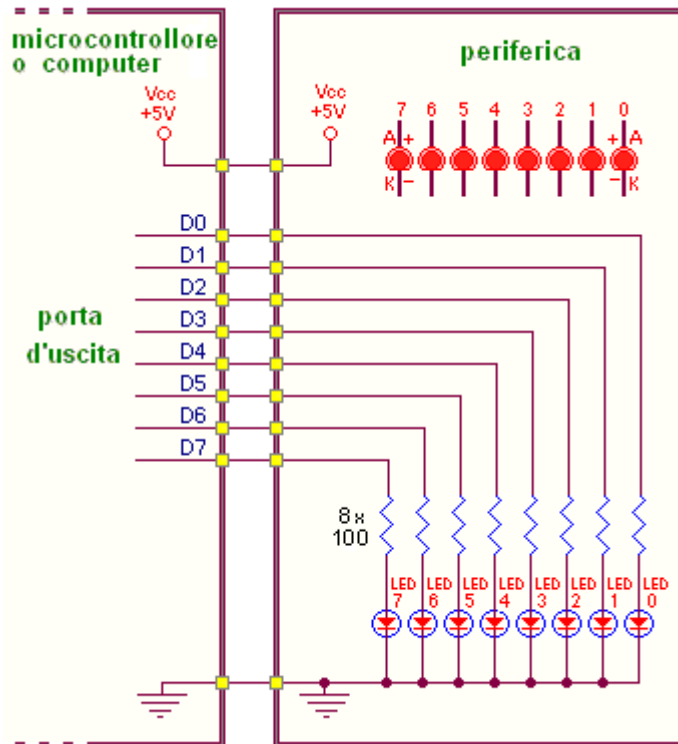
Chi fra noi si è cimentato almeno una volta con i computer, dopo lunghe *piacevoli* (?!) giornate passate a provare (su un simulatore) l'effetto delle sue *istruzioni*, ha certamente coltivato il sogno di *governare il mondo*, prima o poi..

Per tutti, il primo passo verso questo obiettivo è stato quello di collegare un **Led** ad una delle sue *linee d'uscita*, per sottoporlo ad *accensioni* e *spegnimenti* controllati da programma: una emozione pari solo a quella indotta dagli *occhi intriganti* del primo amore..

Immediatamente dopo è sorta l'esigenza di *monitorare* più eventi contemporaneamente, sfruttando al massimo le disponibilità della *porta d'uscita* coinvolta: un **Led** su ciascuna delle *otto linee*; la volta scorsa abbiamo analizzato le caratteristiche delle due soluzioni possibili, auspicando la scelta per quella che chiede alla porta di *assorbire la corrente* necessaria all'accensione di questa *periferica* ...



... piuttosto che sottoporla a inutile *stress*, obbligandola ad *erogarla*:



Il servizio è comunque assicurato, in entrambi i casi.

E' sempre **molto utile** poter disporre di questo circuito: all'inizio nessuno di noi resisterà alla voglia di generare sugli otto punti luminosi gli **effetti più impensabili**, a cominciare dal classico **effetto supercar**, nel quale sono coinvolti solo **due led alla volta**, accesi in sequenza dal centro verso la periferia e viceversa.

Ma, in seguito, questa **periferica** sarà **indispensabile** per collaudare i nostri programmi *importanti*, verificando immediatamente l'attività di un oggetto esterno (una lampada, un motore, un relè, ...) posto sulla stessa **linea d'uscita** senza la necessità di collegarlo (per ora) fisicamente.

Personalmente ritengo *molto utile* creare, per ogni **periferica**, una *tabella riassuntiva* che metta in evidenza, per ogni **dettaglio** del suo schema, il **bit** (della logica di controllo) ad esso associato e il **livello logico** da imporre per ottenere un **effetto attivo** (nel nostro caso *acceso* o *spento*, ma in generale anche qualunque altro *evento fisico binario*).

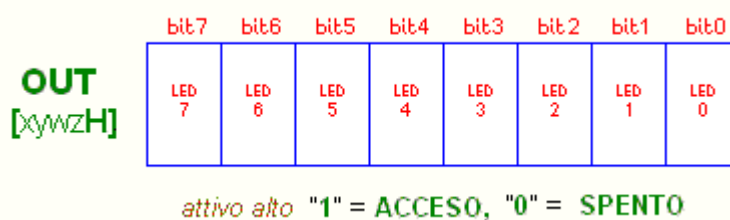
Se il programmatore **non è** la stessa persona che ha progettato il circuito, la disponibilità di quella che amo chiamare **tabella di associazione logica**,

gli consente di disporre immediatamente di ogni dettaglio utile alla **creazione del codice**, senza dover analizzare il circuito ad essa relativo; ma la cosa si dimostra utile e consigliabile anche quando *ciascuno di noi* progetta sia HW che SW, poiché la **stesura del programma** è una attività lontana dal momento della messa a punto del circuito, rendendo spesso necessaria una rapida supervisione per fissare le idee.

Per il progetto della **batteria di 8 Led** che chiede alla porta di **assorbire la corrente** necessaria alla loro accensione vale la seguente **tabella di associazione logica**:



mentre, per quello che chiede alla porta di **erogare la corrente** necessaria, vale:



Per entrambe le **tabelle** (e per ogni altra futura..) viene in sostanza proposta la configurazione del **byte** da porre sulla **porta d'uscita** di indirizzo **OUT [xywzH]**, specificando (per ciascuno dei suoi 8 bit) sia il **livello attivo** che l'**oggetto** ad esso associato; naturalmente il numero binario **xywzH** è l'**indirizzo non ambiguo** della **periferica** da controllare e deve essere sostituito con quello effettivo, che dipende dal tipo di controllore utilizzato. Per esempio, nel caso della **porta parallela** di un **PC** esso sarà: **OUT [0378H]** [vedi <http://www.giobe2000.it/HW/Parallela\index\Progetto05.htm> ].

In conclusione: la disponibilità della **tabella di associazione logica** *virtualizza* la **periferica** rendendo superflua la disponibilità del *suo* schema elettronico nel momento della *sua* programmazione.

### Belli, utili... Ma non basta!

Nessuno potrà fare a meno di una **batteria di 8 led**: costa poco e rende moltissimo; ma i **numeri** sono un'altra cosa!

Fin dalla loro prima apparizione, non solo in ambiente tecnico, hanno appagato il gusto estetico come poche altre cose; cosa sarebbe della *suspense* di un thriller se l'inevitabile (e inesorabile) *bomba* non esibisse il suo contatore con estenuante cadenza?

Mi ricordo l'effetto che mi fecero i quelli strani ed affascinanti *visualizzatori* proposti dal cinema all'inizio (.. ahimè) degli anni sessanta, con i primi *James Bond*: improbabili *tubi di vetro*, piccole valvole con catodi a forma di numero, posti uno dietro l'altro; l'alone luminoso generato dalla scarica sul gas in essi contenuto rimbalzava avanti e indietro, illuminando di rosso la cifra coinvolta, in una danza mai vista prima..

A quei tempi risale il germe della mia malattia, esplosa in tutta la sua virulenza qualche anno dopo: il mio primo progetto fu proprio un "*orologio con i secondi*", un vero lusso per quei tempi, con le sue sei *nixie* fiammeggianti..

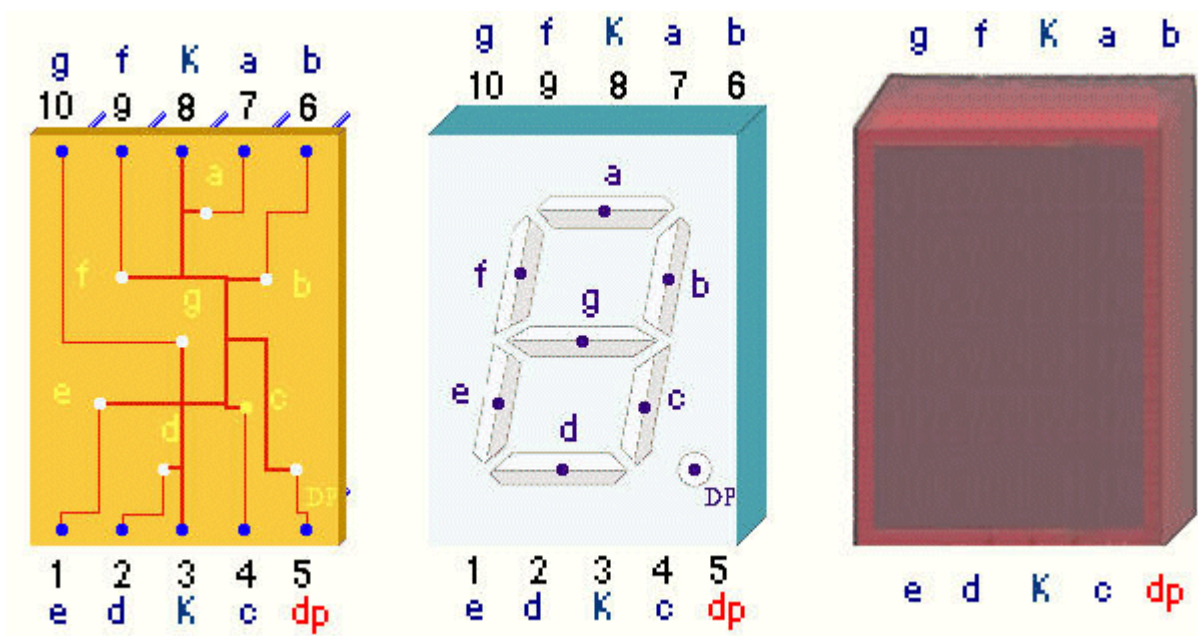
Oggi non esiste nulla che non porti uno *straccio di display*, molto più pratico dei primi che, dovendo disporre per l'accensione delle cifre di una tensione continua di circa **170V** tra *griglia/anodo* ed *catodi*, erano anche pericolosi e delicati. Sebbene ne esistano di diversa natura tecnologica, i più pratici ed economici sono certamente quelli *allo stato solido*, sostanzialmente a **Led**.

Apriamo una parentesi per analizzare la sorprendente *struttura fisica* di un **digit** (= *numero*): esternamente si distinguono chiaramente **sette** piccoli **segmenti sagomati**, raggruppati per comporre un "8" appena evidente sotto la plastica rossa, ma *dentro* è tutta una magia:

- lo strato più basso è, spesso, una piccola *schedina di vetronite* (della dimensione del componente) sulla quale si notano (a fatica) 8

minuscole "*cacchine di mosca*" collegate tra loro da sottili piste che partono e arrivano ai piedini di metallo, posti su due lati opposti; naturalmente si tratta di un piccolo deposito di *semiconduttore* (di solito *arseniuro* o *fosforo di gallio*). Sotto ponendola, con attenzione, alla corretta *differenza di potenziale*, la *piccola massa* si accende, di solito con un'intensità decisamente deludente

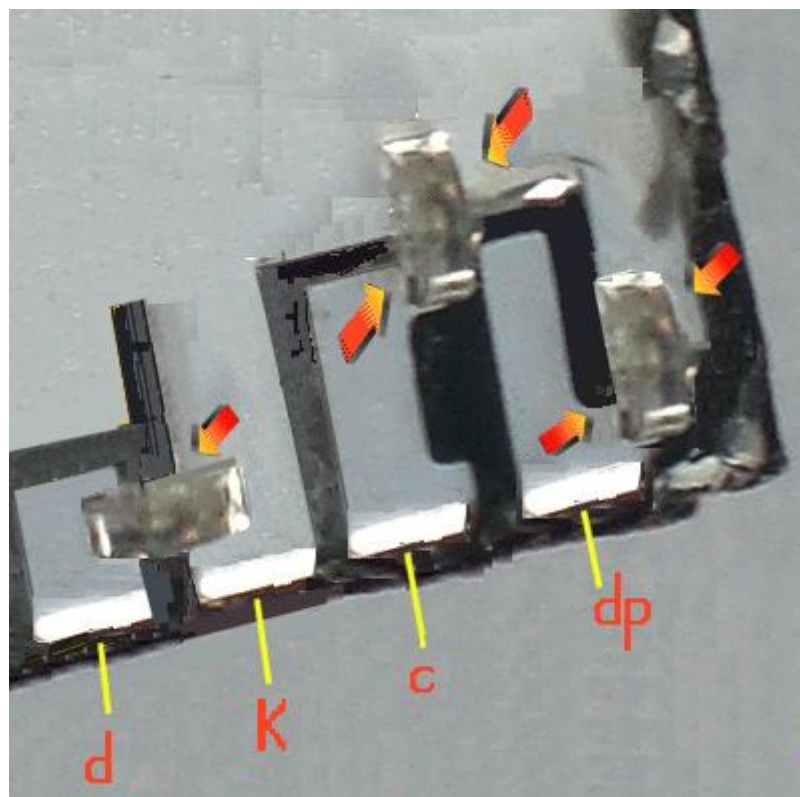
- subito sopra si appoggia un parallelepipedo di plastica opaca, di superficie uguale a quella della schedina e alto 3 o 4 mm; in esso sono incisi dei piccoli solchi a sezione triangolare, con le pareti "a specchio" e un piccolo foro al centro, dal quale emerge il semiconduttore. Applicando di nuovo la corretta *tensione*, la *piccola luce* tende a riflettersi in tutta la dimensione del solco, senza ancora particolare visibilità
- l'ultimo elemento del componente è una scatola di plastica trasparente rossa, chiamata a contenere le due parti precedenti; il suo compito però non si limita a questo: la superficie interna del suo fondo è coperta di minuscole piramidi a sezione quadrata, in pratica un *catarifrangente* come quelli delle luci posteriori dei nostri veicoli. Poiché la sua *arte* è quella di riflettere la luce nella stessa direzione da cui è scaturita, le infinite riflessioni sulla sottostante *guida ottica riflettente* si estendono in un attimo a tutta la dimensione del solco, *integrando* e *filtrando* la luce e facendole assumere la dignità di *segmento*



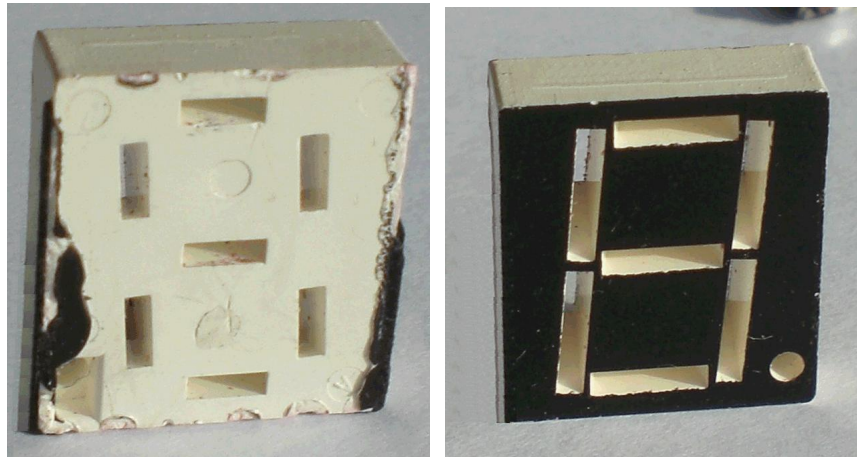


Uno dei più diffusi sul mercato è il **digit FND500**: esso rispetta la logica costruttiva appena descritta ma offre una soluzione tecnologica più pregiata:

- ora la *schedina di vetronite* è sostituita da un agglomerato di plastica nera (della stessa dimensione) nella quale è *affogata* una serie di robusti lamierini *sagomati* con la forma dei collegamenti interni, in parte *terminati* con i dieci piedini da esso emergenti, ovviamente ad angolo retto, rispetto ai collegamenti stessi
- ciascuno degli 8 *elementi fotoemittenti* è ora *fisicamente* un *microled*, simile ad un mezzo disco di plastica trasparente, di 3 mm di diametro e spesso 1 millimetro, col chiaro compito di fare da *lente* al semiconduttore posto esattamente al centro della sua base. Il dettaglio fotografico mostra l'angolo in basso a destra della resina nera: si distinguono quattro lamierini che (sul lato inferiore) continueranno verso l'esterno sottoforma di piedini (**pin5 = decimal point**, **pin4 = segmento c**, **pin3 = catodo**, **pin2 = segmento d**); il più grande è ovviamente quello associato al **catodo comune** e, da esso verso gli altri, si notano i tre *microled* relativi agli elementi del **digit** appena citati:



- la guida ottica centrale, molto più *spartana* di quella descritta in precedenza, da un lato accoglie ora i *microled*, offrendo fessure 3x1 mm esattamente adatte ad ospitarli, e, dal lato opposto, porta le fessure alla dimensione finale di ciascun segmento, 6x1,5 mm

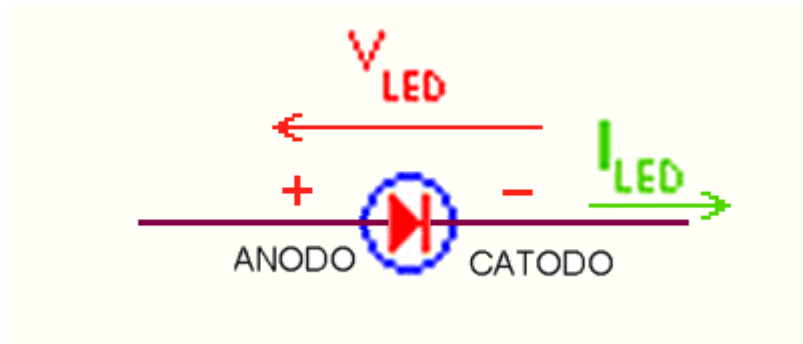


- l'*effetto lente* garantito dalla struttura dei *microled* rende inutile la presenza del *catarifrangente* sulla superficie interna dell'involucro di plastica rossa, che risulta pertanto liscio e trasparente; esso mantiene comunque il compito di *esaltare* la componente rossa della luce emessa, garantendo in fine un eccellente risultato

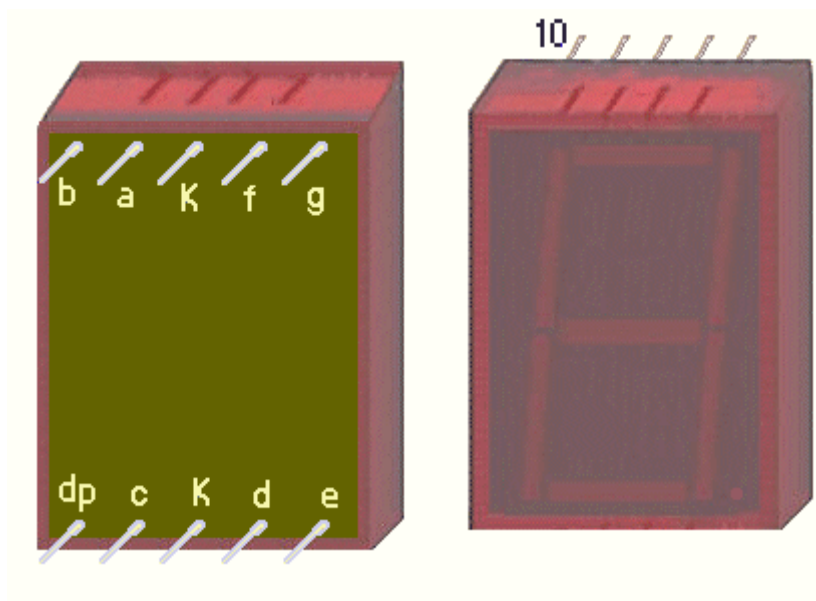
I **digit a Led** sono disponibili in due forme **funzionali alternative**, dette a **catodo comune** e ad **anodo comune**; noi tutti sappiamo che i *diodi a semiconduttore* consentono il passaggio di una *corrente diretta* se sottoposti ad una corretta *differenza di potenziale*, a partire dalla *tensione di soglia*, discussa la volta scorsa...

Di certo la *corrente convenzionale* scorre dal polo **positivo** verso quello **negativo** del generatore che la produce e il *simbolo* del diodo è stato pensato per evidenziare questo evento; la sua forma a *triangolo* può essere presa come regola per ricordare il senso di percorrenza della *corrente convenzionale*. Va solo ricordato che il lato in cui essa entra è detto **anodo** (ed è sottoposto al polo positivo della *tensione*) mentre quello da cui essa esce è detto **catodo** (ed è collegato al polo negativo della *tensione*):



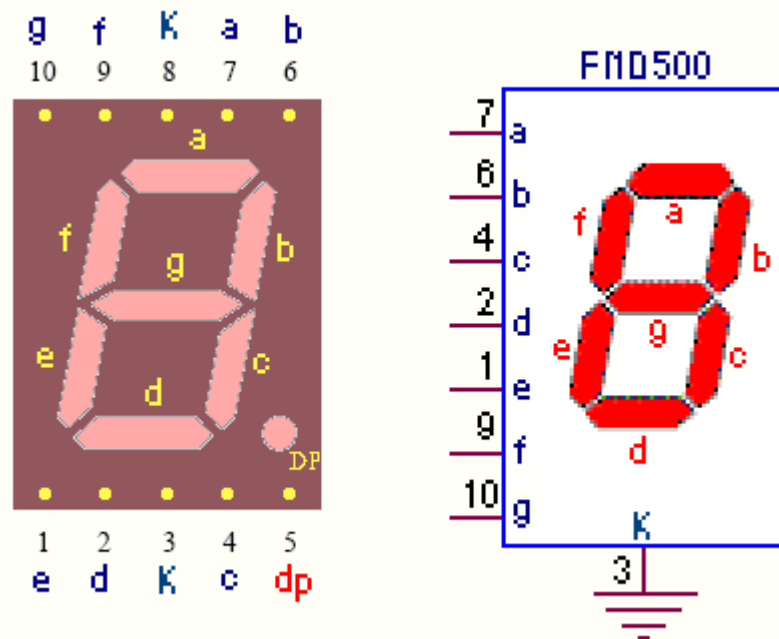


Uno dei problemi più sentiti, quando si montano questi dispositivi è quello di ricordare la *pedinatura*: quando serve *d'urgenza*, non si riesce mai a trovarla... La figura seguente ne riassume l'aspetto, sia per i **digit a anodo comune**, **FND507**, che per quelli a **catodo comune**, **FND500**; è interessante notare che la disposizione delle lettere sulla periferia del componente è solo apparentemente illogica: si tratta della sequenza che garantisce il collegamento a ciascuno dei *led interni* con il percorso più breve possibile, come si può facilmente verificare osservando i dettagli proposti in precedenza.



Anche la disponibilità di *due catodi* (o di *due anodi*, sul **pin3** e sul **pin8**) non deve stupire; basta pensare alla difficoltà nella creazione di *circuiti stampati* con molti **digit** (di solito multipli di 2): poter disporre di due possibilità, di qua o al di là di numerose *fasce di piste*, è certamente un vantaggio.

Infine, data la sua simmetria, nella *messa in esercizio* può tornare utile sottolineare che solo uno dei suoi due *lati corti* è caratterizzato dalla presenza di *quattro visibilissime tacchette*: il **pin10** di questo componente è il primo a sinistra rispetto a questo lato. Chi dispone di buona vista può arrivare alla stessa conclusione anche individuando il puntino nell'angolo in basso a destra...



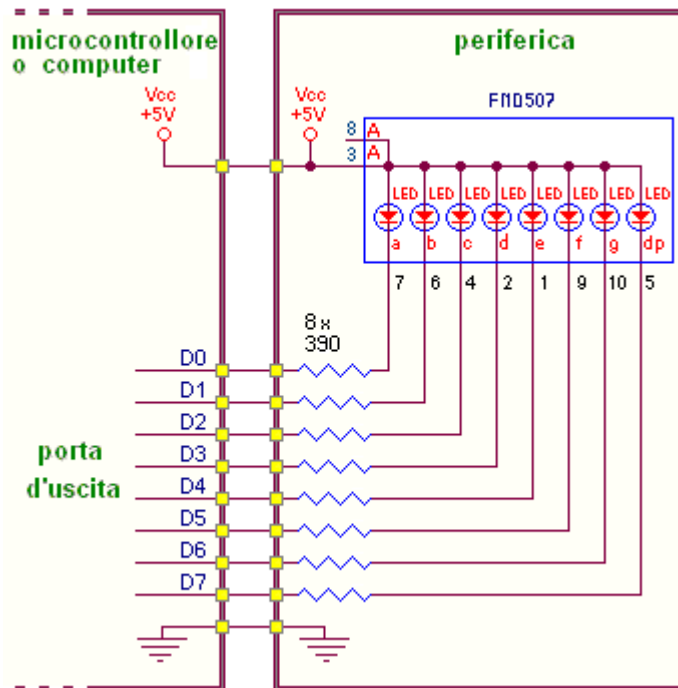
Non di rado questa *piedinatura* è mantenuta inalterata anche su *modelli alternativi*; di solito il progettista si limita a scegliere questa **periferica optoelettronica**, tra i numerosi modelli offerti sui banchi dei rivenditori, in funzione della sua *forma estetica* o delle sue *necessità d'ingombro*, non curandosi (come succede per altri componenti) di conoscere il suo produttore; per questo le sigle dei **digit** proposti come esempio sono sostanzialmente irrilevanti:



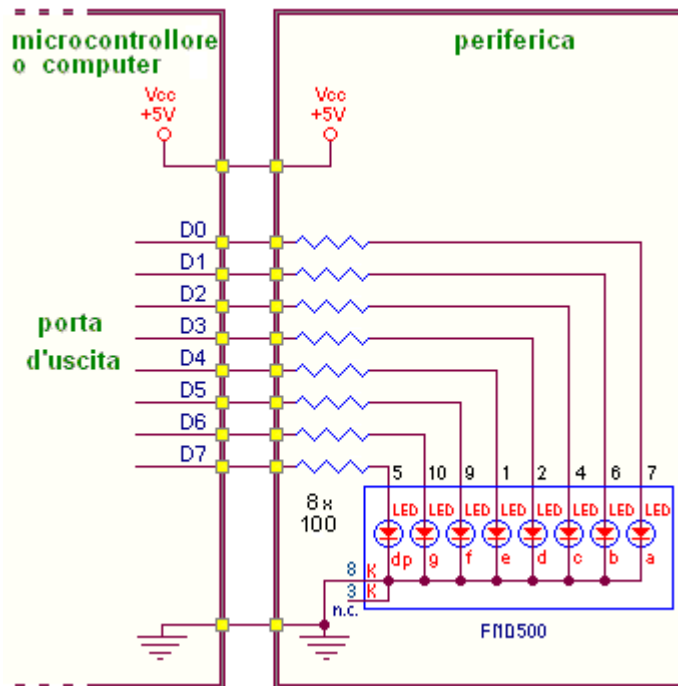
Sebbene la natura di questo componente sia sostanzialmente quella dei led che lo compongono può essere istruttivo consultare le pagine dei **data sheet originali**; poichè essi sono di difficile reperibilità ho provveduto ad allegarne alcuni: [FND500\\_1.pdf](#) , [FND500\\_2.pdf](#) e [tdsl5160.pdf](#)

Dunque: nessun appassionato di computer può ignorare il fascino di qualche **digit** da pilotare.. L'utilizzo diretto di un **digit** (senza *strato* di **interfaccia**) sulle **porte d'uscita** dei dispositivi programmabili (*microcontrollori* o *PC*) è ancora legittimo, riflettendo le considerazioni *di carico* fatte all'inizio per le **batterie di 8 led**, alle quali ci si può inesorabilmente ricondurre.

Con tutti i dispositivi che esercitano un controllo logico *attivo basso* sono necessari **digit** ad **anodo comune**, come gli **FND507**: ciascuna delle **otto linee** delle **porte d'uscita** è chiamata ad **assorbire** la corrente necessaria all'accensione dei rispettivi led di questa **periferica**:





Analogamente i **digit a catodo comune**, come gli **FND500**, sono necessari quando il controllo logico operato dalle porte d'uscita dei dispositivi programmabili (*micro* o *PC*) è *attivo alto*, costringendo ciascuna delle **otto linee** ad **erogare la corrente** necessaria, come noto condizione *non ideale* ma molto più diffusa della precedente, probabilmente (come già sottolineato) per il fatto di **non** dover disporre di *alimentazione esterna*, non sempre disponibile:



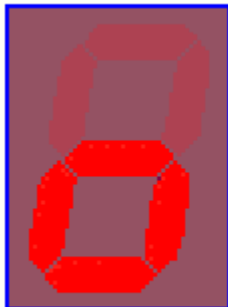

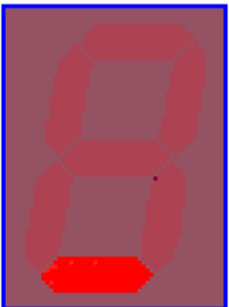
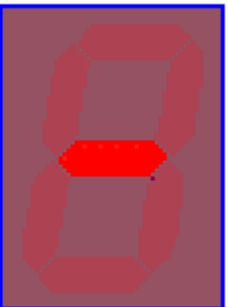

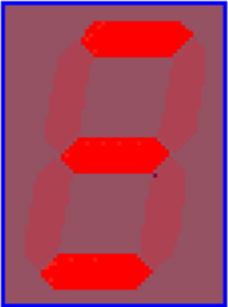

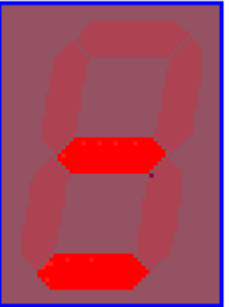


Naturalmente, in entrambi i casi, non faremo mancare un **resistore** in serie ad ogni **Led** (*segmento* interno), il cui valore di *resistenza* dovrà assicurare una corrente nei limiti dello standard, in accordo con le considerazioni e i calcoli sviluppati nella precedente puntata.

Le **tabelle di associazione logica** chiamate a *virtualizzare* le due soluzioni proposte sono le seguenti:

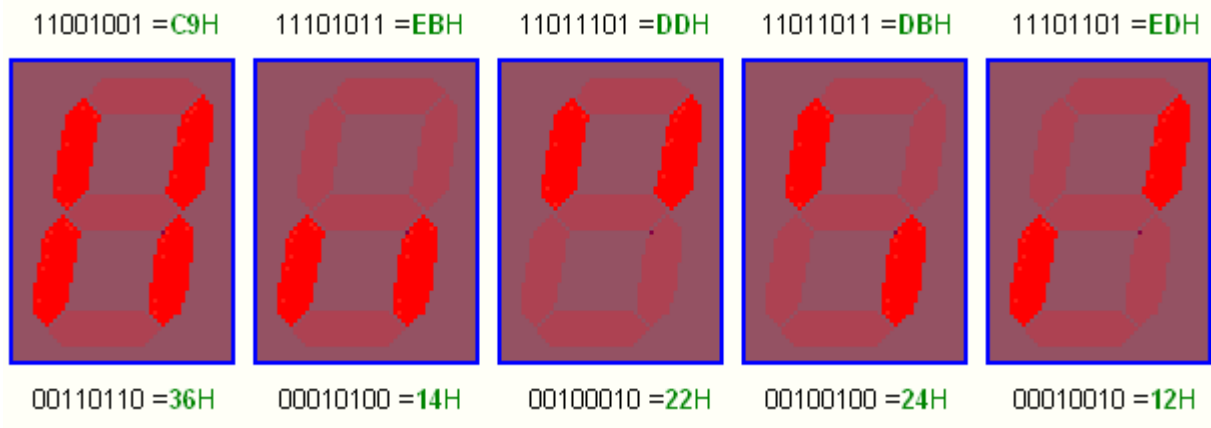
| <p><b>digit FND507</b><br/>ad <b>anodo comune</b><br/>la porta d'uscita <b>assorbe</b> corrente</p> | <p><b>OUT</b><br/>[xywzH]</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>bit7</th> <th>bit6</th> <th>bit5</th> <th>bit4</th> <th>bit3</th> <th>bit2</th> <th>bit1</th> <th>bit0</th> </tr> </thead> <tbody> <tr> <td>POINT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> </tr> <tr> <td>dp</td> <td>g</td> <td>f</td> <td>e</td> <td>d</td> <td>c</td> <td>b</td> <td>a</td> </tr> </tbody> </table> <p style="text-align: center;"><i>attivo basso</i> "0" = <b>ACCESO</b>, "1" = <b>SPENTO</b></p>  | bit7    | bit6    | bit5    | bit4    | bit3    | bit2    | bit1 | bit0 | POINT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | dp | g | f | e | d | c | b | a |
|---|--|---------|---------|---------|---------|---------|---------|------|------|-------|---------|---------|---------|---------|---------|---------|---------|----|---|---|---|---|---|---|---|
| bit7  | bit6   | bit5    | bit4    | bit3    | bit2    | bit1    | bit0    |      |      |       |         |         |         |         |         |         |         |    |   |   |   |   |   |   |   |
| POINT   | SEGMENT  | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT |      |      |       |         |         |         |         |         |         |         |    |   |   |   |   |   |   |   |
| dp  | g  | f       | e       | d       | c       | b       | a       |      |      |       |         |         |         |         |         |         |         |    |   |   |   |   |   |   |   |
| <p><b>digit FND500</b><br/>a <b>catodo comune</b><br/>la porta d'uscita <b>eroga</b> corrente</p>   | <p><b>OUT</b><br/>[xywzH]</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>bit7</th> <th>bit6</th> <th>bit5</th> <th>bit4</th> <th>bit3</th> <th>bit2</th> <th>bit1</th> <th>bit0</th> </tr> </thead> <tbody> <tr> <td>POINT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> <td>SEGMENT</td> </tr> <tr> <td>dp</td> <td>g</td> <td>f</td> <td>e</td> <td>d</td> <td>c</td> <td>b</td> <td>a</td> </tr> </tbody> </table> <p style="text-align: center;"><i>attivo alto</i> "1" = <b>ACCESO</b>, "0" = <b>SPENTO</b></p>   | bit7    | bit6    | bit5    | bit4    | bit3    | bit2    | bit1 | bit0 | POINT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | dp | g | f | e | d | c | b | a |
| bit7  | bit6   | bit5    | bit4    | bit3    | bit2    | bit1    | bit0    |      |      |       |         |         |         |         |         |         |         |    |   |   |   |   |   |   |   |
| POINT   | SEGMENT  | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT | SEGMENT |      |      |       |         |         |         |         |         |         |         |    |   |   |   |   |   |   |   |
| dp  | g  | f       | e       | d       | c       | b       | a       |      |      |       |         |         |         |         |         |         |         |    |   |   |   |   |   |   |   |

Sebbene un **digit** sia sostanzialmente una **batteria di 8 led**, la sua filosofia d'uso è completamente diversa, rivolta a **mostrare** una qualunque combinazione dei suoi **sette segmenti**; il suo **collegamento diretto** su di una **porta d'uscita** (rispetto ad altre soluzioni, che tratteremo più avanti) si presta ad alcune interessanti considerazioni.

In **primo luogo** c'è il **vantaggio** del controllo assoluto sui **caratteri proponibili** su di esso: con le opportune combinazioni di **bit attivi** è possibile proporre **qualunque simbolo**, non solo alfanumerico, consentendo perfino di tenere **spento** il **digit**:

|   |   |   |  |   |
|---|---|---|--|---|
| 10100011 = <b>A3H</b>   | 10011100 = <b>9CH</b>   | 11110111 = <b>F7H</b>   | 10111111 = <b>BFH</b>  | 11111110 = <b>FEH</b>   |
|   |   |   |   |   |
| 01011100 = <b>3EH</b>   | 01100011 = <b>63H</b>   | 00001000 = <b>08H</b>   | 01000000 = <b>40H</b>  | 00000001 = <b>01H</b>   |
| 10110110 = <b>B6H</b>   | 11110110 = <b>F6H</b>   | 10110111 = <b>B7H</b>   | 10111110 = <b>BEH</b>  | 11111111 = <b>FFH</b>   |
|  |  |  |  |  |
| 01001001 = <b>49H</b>   | 00001001 = <b>09H</b>   | 01001000 = <b>48H</b>   | 01000001 = <b>41H</b>  | 00000000 = <b>00H</b>   |





Sebbene non strettamente necessarie, queste *combinazioni di segmenti* possono rivelarsi molto utili per fornire particolari segnalazioni, come *allarmi* o *segnali di attesa* e non saranno disponibili con altre soluzioni; ognuno dei quindici esempi offre il valore del **byte** necessario per ottenerlo, ottenuto proprio con l'aiuto delle rispettive **tabelle di associazione logica** :

- sul lato superiore per i **digit FND507** ad **anodo comune**, sui quali è esercitato un controllo logico *attivo basso*
- sul lato inferiore per i **digit FND500** a **catodo comune**, sui quali è esercitato un controllo logico *attivo alto*

Da notare che il **bit più significativo (bit7)** di tutte le combinazioni è sempre disattivo, essendo associato al *Decimal Point*, mai coinvolto negli esempi; per altro le due soluzioni proposte rendono agevole anche la sua gestione consentendo di:

- accenderlo con 01111111 = **7FH** e spegnerlo con 11111111 = **FFH**, in logica *attiva bassa*
- accenderlo con 10000000 = **80H** e spegnerlo con 00000000 = **00H**, in logica *attiva alta*

La **seconda considerazione** riguarda la gestione dei *caratteri alfanumerici*, compito tipico delle applicazioni che richiedono una **periferica** come questa: essa è **resa** relativamente **difficile**, essendo necessario risalire di volta in volta al valore (**byte**) binario corrispondente al simbolo desiderato; per renderla agevole è opportuna la creazione e la disponibilità di **tabelle di conversione** da gestire da software; di certo il problema è irrisorio per un bravo programmatore e il risparmio di hardware può indurre in tentazione...

Ecco il valore del **byte** necessario per ottenere i sedici simboli del *sistema di numerazione esadecimale* ([http://www.giobe2000.it/Tutorial/Schede/08-Misura\\_Informazione/809.htm](http://www.giobe2000.it/Tutorial/Schede/08-Misura_Informazione/809.htm)), ottenuti ancora con l'aiuto delle **tabelle di associazione logica**, rispettivamente per logiche *attive basse* (**FND507**) e per logiche *attive alte* (**FND500**)

|   | digit <b>FND507</b><br>ad anodo comune | digit <b>FND500</b><br>a catodo comune |
|---|--|--|
| 0 | 11000000 = C0H                         | 00111111 = 3FH                         |
| 1 | 11111001 = F9H                         | 00000110 = 06H                         |
| 2 | 10100100 = A4H                         | 01011011 = 5BH                         |
| 3 | 10110000 = B0H                         | 01001111 = 4FH                         |
| 4 | 10011001 = 99H                         | 01100110 = 66H                         |
| 5 | 10010010 = 92H                         | 01101101 = 6DH                         |
| 6 | 10000010 = 82H                         | 01111101 = 7DH                         |
| 7 | 10000111 = 87H                         | 01111000 = 78H                         |
| 8 | 10000000 = 80H                         | 01111111 = 7FH                         |
| 9 | 10010000 = 90H                         | 01101111 = 6FH                         |
| A | 10001000 = 88H                         | 01110111 = 77H                         |
| b | 10000011 = 83H                         | 01111100 = 7CH                         |
| c | 10100111 = A7H                         | 01011000 = 58H                         |
| d | 10100001 = A1H                         | 01011110 = 5EH                         |
| E | 10000110 = 86H                         | 01111001 = 79H                         |
| F | 10001110 = 8EH                         | 01110001 = 71H                         |

Nella prossima puntata riprenderemo ad occuparci di **digit**, ma da un punto di vista diverso; i dispositivi programmabili (*microcontrollore* o *personal computer*) potranno fruire di *strutture specializzate* alle quali delegare alcuni dei compiti descritti in questa lezione..

---

Questo Tutorial è **del tutto originale**, creato e pensato per gli amici di **Grix** e articolato in numerose *puntate*...

Ogni suo **testo, immagine, schema, progetto** è protetto dalla normativa sul **diritto d'autore** [[http://www.siae.it/Faq\\_siae.asp](http://www.siae.it/Faq_siae.asp)]; la **riproduzione a fini commerciali**, totale o parziale, è quindi **vietata** in qualunque forma, su qualsiasi supporto e con qualunque mezzo.

L'autore è invece orgoglioso di offrire gratuitamente il suo lavoro e la sua esperienza a chiunque desidera arricchire le proprie conoscenze, autorizzando la **stampa** di quest'opera per un uso **personale e non commerciale** e l'eventuale utilizzo dei contenuti in ambiti esclusivamente amatoriali o didattici con la *speranza* che ne venga almeno citata la fonte.

Puoi [scaricare qui la versione PDF](#) della **TERZA PARTE**

Alcuni argomenti **citati** in questa parte sono **disponibili** sul sito



una ricca raccolta di dispositivi e progetti pensata per aiutarti a comprendere i segreti del tuo computer [sia *Personal* che *microcontrollore*]

In particolare è **consigliata** la lettura di

[Tutorial Scheda 08 - La Misura dell'Informazione](#)

**Come si misura l'informazione?** - *Bit, Nibble, Byte, Word, DoubleWord*  
**I multipli del Byte** - *Le potenze di 2, KiloBytes, MegaBytes, GigaBytes*  
**Il sistema di numerazione decimale** - *Elementi e Base di un sistema di Numerazione*  
**Il sistema di numerazione binario** - *da Binario a Decimale*  
**Il sistema di numerazione esadecimale** - *i 16 Elementi, da Esadecimale a Decimale*  
**Le conversioni inverse** - *da Decimale a Binario, da Decimale a Esadecimale*

[Progetti - Output Dati a 8 bit su porta Parallela](#)

OUTPUT **dati** a 8 bit su **SPP** verso 0378H - *codici Ascii su batteria di 8 led*  
OUTPUT **dati** a 8 bit su **SPP** verso 0378H - *effetto SuperCar su batteria di 8 led*