

# SISTEMI ELETTRONICI AUTOMATICI

TESINA ESAMI DI STATO – 2003

CASAMASSIMA NICOLA – 5ETB

ISTITUTO TECNICO INDUSTRIALE "M. PANETTI"- BARI

Docente: Prof. Ettore Panella

## CODIFICATORE CON PRIORITA'

Si riporta la realizzazione di un circuito **codificatore con priorità** in grado di visualizzare, su un display a sette segmenti a catodo comune, il numero corrispondente, all'ingresso attivato, tramite microswitch ad 8 microinterruttori, avente maggior priorità. Il sistema è gestito da software in Visual Basic 6.

Il sistema codificatore è stato realizzato sfruttando l'interfaccia parallela Centronics secondo lo schema mostrato in figura 1. Per l'acquisizione di un byte si utilizza il 74LS244 che contiene 8 buffer tri-state utilizzato come multiplexer che consente il passaggio alternato di due coppie di 4 bit. Il computer oltre a svolgere la funzione di codificatore, si comporta anche da decodificatore per il pilotaggio del display.

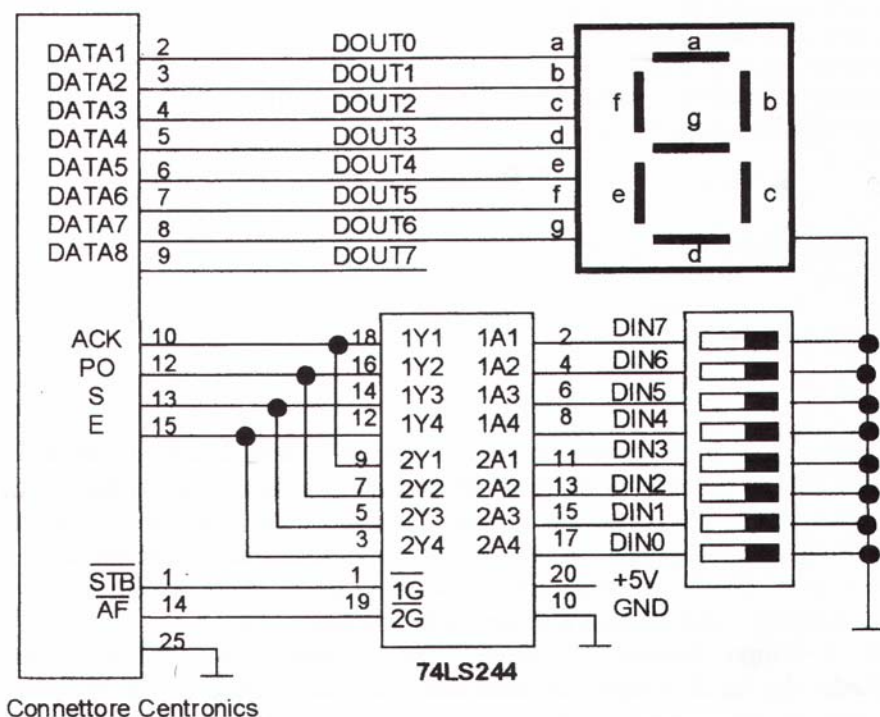
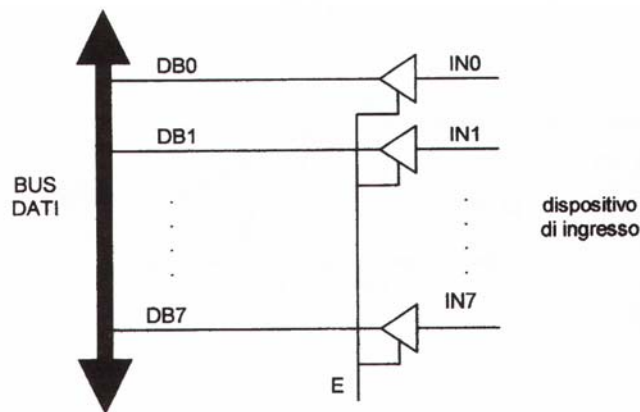


Figura 1 – Circuito di interfaccia parallela che utilizza il registro di stato dell' interfaccia Centronics per l'acquisizione del dato digitale.

Di seguito si è prodotto un programma in Visual Basic 6 in grado di acquisire il dato di ingresso con priorità e di decodificarlo per la visualizzazione sul display a sette segmenti a catodo comune.

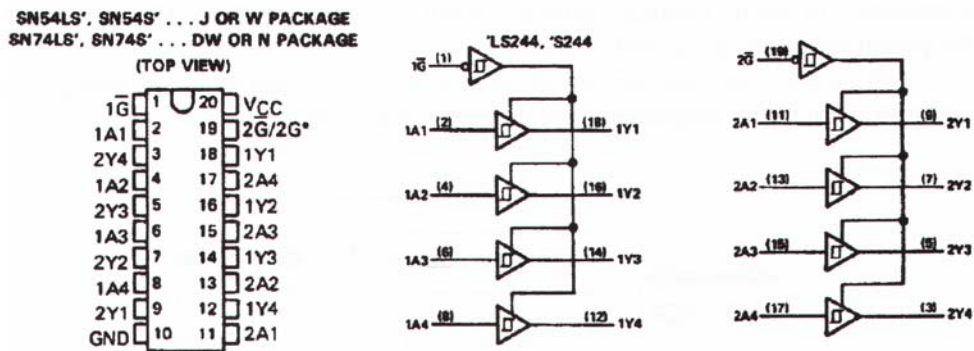
### Buffer TRI – STATE

Il collegamento di un dispositivo periferico di input al bus dati di un sistema a microprocessore deve avvenire interponendo dei buffer tri-state non invertenti. L'ingresso di abilitazione di ciascun buffer consente il collegamento dei dati del periferico di input al bus dei dati del microprocessore. Ponendo l'ingresso di abilitazione al livello logico opposto i buffer portano le loro uscite nello stato tri-state e quindi, in pratica, isolano elettricamente il dispositivo di input dal bus. In figura 2 si mostra un tipico collegamento.



**Figura 2 – Collegamento di un dispositivo di ingresso al BUS dati di un microprocessore.**

Per questo tipo di collegamento è sufficiente usare l'integrato 74LS244 che contiene 2 gruppi indipendenti di 4 buffer tri-state a trigger di Schmitt. Se gli ingressi di abilitazione di ciascun gruppo: 1G e 2G sono tenuti al livello alto l'uscita di ciascun buffer ha lo stesso livello logico del proprio ingresso; se, invece gli ingressi di abilitazione sono al livello basso, le uscite del buffer si portano in tri-state ( stato di alta impedenza) indipendentemente dal livello logico applicato in ingresso. Si riporta in figura 3, la piedinatura e lo schema interno dell'integrato 74LS244 con i relativi data sheet.

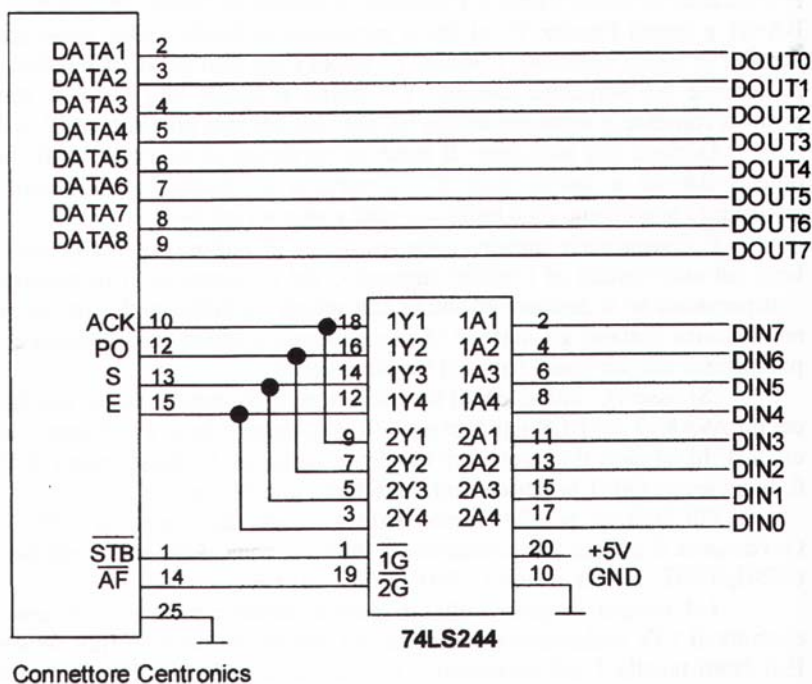


**Figura 3 – Piedinatura e schema interno del 74LS244.**

## Interfaccia parallela esterna per l'I/O digitale.

Una possibile soluzione per interfacciare circuiti digitali con 8 bit di ingresso ed 8 bit di uscita al personal computer è quella di utilizzare l'interfaccia standard Centronics già disponibile sul retro del PC e descritta di seguito. Poiché l'interfaccia Centronics fornisce un byte di uscita ma accetta solo cinque bit in ingresso si ricorre, per l'acquisizione di un byte, alla tecnica del multiplex facendo acquisire al PC prima 4 bit e successivamente gli altri 4 bit del byte. L'acquisizione di un byte, per tanto, avverrà attraverso due fasi successive di acquisizione a quattro bit. Il software di gestione provvederà alla opportuna fusione dei due semibyte.

In figura 4 si riporta uno schema elettrico molto semplice che impiega solamente un 74LS244 che, come è noto, contiene due gruppi indipendenti di buffer tri-state.



**Figura 4 – Interfaccia parallela esterna per l'I/O digitale.**

Supponendo di utilizzare il linguaggio Visual Basic, l'algoritmo di acquisizione del byte applicato alle linee DIN7...DINO è il seguente:

<b>OUT 890, 1</b>	Generazione del livello basso sul bit $\overline{STB}$ ed alto sul bit $\overline{AF}$ per cui i bit 1Y1 1Y2 1Y3 1Y4 sono abilitati mentre i bit 2Y1 2Y2 2Y3 2Y4 sono in tri-state.
<b>AH = (INP(889) AND 120) * 2</b>	Acquisizione e sistemazione dei 4 bit di ingresso più significativi nella parte alta del byte A.
<b>OUT 890, 2</b>	Generazione del livello alto sul bit $\overline{STB}$ e basso sul bit $\overline{AF}$ per cui i bit 1Y1 1Y2 1Y3 1Y4 sono in tri-state mentre i bit 2Y1 2Y2 2Y3 2Y4 sono abilitati.
<b>AL = (INP(889) AND 120) / 8</b>	Acquisizione e sistemazione dei 4 bit di ingresso meno significativi nella parte bassa del byte A.
<b>A = AH + AL</b>	Composizione del byte A.
<b>OUT 890, 0</b>	Generazione del livello alto sia su $\overline{STB}$ che su $\overline{AF}$ per cui sono in tri-state i bit 1Y1 1Y2 1Y3 1Y4 ed i bit 2Y1 2Y2 2Y3 2Y4.

L'AND con 120 (01111000) consente di estrarre solo i quattro bit utilizzati dall' interfaccia forzando gli altri a 0. La moltiplicazione per due produce lo scorrimento a sinistra di un bit mentre la divisione per 8 produce lo scorrimento a destra di 3 bit. Pertanto si ha :

AH=	DIN7	DIN6	DIN5	DIN4	0	0	0	0
AL=	0	0	0	0	DIN3	DIN2	DIN1	DIN0
A=AH+AL	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0

L'algoritmo per la generazione del byte B di uscita sulle linee DOUT7...DOUT0 è:

### OUT 888, B

Per poter funzionare correttamente, il 74LS244 deve essere alimentato con una tensione costante di +5V prelevata da una alimentatore esterno.

### Interfacce standard

Le interfacce standard sono quelle che rendono compatibile il collegamento tra un dispositivo periferico ed il computer.

Le interfacce standard, normalmente, sono costituite da schede elettroniche inserite in una slot di espansione di un computer: ISA, PCI, AGP. In quest'ultima, in realtà, può trovar posto solo una scheda grafica in grado di pilotare un monitor attraverso il connettore VGA disponibile sul retro del PC. Le schede grafiche più recenti presentano anche una uscita TV in grado di comandare un normale ricevitore televisivo oppure un videoregistratore.

Le interfacce standard su schede più recenti sono disponibili solo su connettori PCI.

Le interfacce standard più note sono:

1. Interfaccia Centronics
2. Interfaccia SCSI
3. Interfaccia RS-232
4. Interfaccia USB
5. Interfaccia IEEE1394

La prima è un'interfaccia parallela a 8 bit. Il suo nome è quello della ditta di stampanti che per prima diffuse questa interfaccia per poter, appunto, collegare una stampante al computer. Utilizzata su ogni computer per oltre un ventennio per connettere stampanti e dispositivi periferici come scanner, webcam, ecc., è attualmente in disuso a causa della limitata velocità di funzionamento.

L'interfaccia SCSI (Small Computer System Interface), anch'essa parallela ad 8 bit e ratificata nel 1986, ha subito nel tempo numerose evoluzioni ed è utilizzata per collegare dispositivi destinati ad essere particolarmente veloci: hard-disk, scanner, masterizzatori.

L'interfaccia RS-232 è di tipo seriale, ha subito nel tempo delle evoluzioni che ne hanno consentito di aumentare la velocità di trasferimento dati che, comunque, rimane abbastanza basso (256Kbit/s.) e ridotto il numero delle linee effettivamente utilizzate.

L'interfaccia USB (Universal Serial Bus), apparsa nel 1995, è destinata a soppiantare la RS-232 per alcune altre notevoli caratteristiche e per la sua elevata velocità che può raggiungere il valore di 12Mbit/s e di 420Mbps per la versione 2.0.

L'interfaccia IEEE1394, anch'essa seriale, consente un flusso massimo di dati pari a 400Mbps con la possibilità di collegare fino a 63 dispositivi in serie o a stella.

L'inserzione di nuovi dispositivi collegati a tale interfaccia può avvenire a caldo.

Sulla maggior parte delle moderne schede madri di PC sono già presenti le interfacce Centronics, RS-232 e USB. In tal modo restano disponibili i connettori ISA e PCI per altre interfacce standard e non standard.

Alcune schede madri, oltre a contenere le citate interfacce, contengono anche alcune altre interfacce, normalmente presenti su slot e, addirittura esterne. Tali schede madri sono denominate "all-in-one" e contengono:

- 1 Centronics;
- 2 Seriali RS-232;
- 2 USB;
- 1 Modem interno;
- 1 interfaccia grafica;
- 1 interfaccia audio.

Tale soluzione risulta particolarmente economica ed interessante ma poco si adatta ad esigenze legate all'aggiornamento delle citate interfacce già integrate su scheda.

### Interfaccia Centronics

E' una interfaccia parallela ad 8 bit di tipo asincrona usata soprattutto per collegare un computer ad una stampante parallela. Il flusso di dati è tipicamente monodirezionale e va, naturalmente, dal computer alla stampante. Recentemente, grazie alla evoluzione della interfaccia, è possibile anche inviare sulle stesse linee dati che vanno dal dispositivo periferico al computer e ciò consente l'utilizzo di tale interfaccia anche per il collegamento di dispositivi di input come, ad esempio, lo scanner o la webcam. Il connettore sul retro del computer è di tipo D a 25 poli femmina. Su un PC possono prendere posto fino a 3 interfacce parallele denominate LPT1, LPT2 e LPT3 (Line Printer Terminal). Ciascuna delle 3 LPT presenta 3 indirizzi contigui destinate alle periferiche di I/O.

L'indirizzo base della LPT1 è 888 (in esadecimale: 378), gli altri due indirizzi sono 889 e 890. L'indirizzo base della LPT2 è 632 (in esadecimale: 278), gli altri due indirizzi sono 633 e 634. L'indirizzo base della LPT3 è 956 (in esadecimale: 3BC), gli altri due indirizzi sono 957 e 958. Si riassume la situazione nella seguente tabella

	Ind. base	Ind. bas e+1	Ind. bas e+2
LPT1	888	889	890
LPT2	632	633	634
LPT3	956	957	958

Alla LPT1 viene riservato l'interrupt IRQ7. Il registro base di indirizzo 888, denominato *registro dati*, contiene 8 bit di uscita dal PC.

Il registro di indirizzo successivo 889, noto come *registro di stato*, è accessibile solo dall'esterno e solamente per 5 dei suoi 8 bit. È detto registro di stato perché ciascuna delle 5 linee individua un particolare stato in cui si trova la stampante (occupata, carta esaurita, errore, ecc.). Il registro di indirizzo 890, noto come *registro di controllo*, rende disponibili solo 4 bit di uscita. In figura 5 si mostra il connettore a 25 poli posto sul retro di un PC.



**Figura 5 – Interfaccia Centronics. Connettore a 25 poli femmina posto sul retro di un computer.**

Si riporta la tabella con la piedinatura, la denominazione, la direzione, l'indirizzo e l'uso delle linee che l'interfaccia Centronics mette a disposizione sul connettore D a 25 poli agli indirizzi 888, 889 e 890 della LPT1.

PIN	NOME	DIREZIONE	INDIRIZZO ED USO
1	$\overline{\text{STROBE}}$	USCITA	890 OUT 890,1 0V 890 OUT 890,0 5V
2	DATA1	USCITA	
3	DATA2	USCITA	
4	DATA3	USCITA	
5	DATA4	USCITA OUT	888,N
6	DATA5	USCITA	Per
7	DATA6	USCITA OUT	N 255
8	DATA7	USCITA	
9	DATA8	USCITA	
10	ACK	INGRESSO	B3 (64) SE ALTO
11	$\overline{\text{BUSY}}$	INGRESSO	B4 (128) SE BASSO
12	PAPER OUT	INGRESSO	B2 (32) SE ALTO
13	SELECTE D	INGRESSO	B1 (16) SE ALTO
14	$\overline{\text{AUTOFEED}}$	USCITA	890 OUT 890,2 0V
15	ERROR	INGRESSO	B0 ( 8 ) SE ALTO
16	INIZIALIZE PRINTER	USCITA	890 OUT 890,4 5V
17	$\overline{\text{SELECT INPUT}}$	USCITA	890 OUT 890,8 0V
18...25	GND		

Le linee di uscita DATA8...DATA1, di indirizzo 888 (378 esadecimale), situate tra i pin 9...2, sono memorizzate, cioè rappresentano i bit di uscita di altrettanti flip-flop.

LINEE	DATA8	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1
PIN	9	8	7	6	5	4	3	2

### **Registro di stato (indirizzo 889):**

L'interfaccia Centronics possiede 5 linee di ingresso all'indirizzo 889 (379 esadecimale) con i seguenti valori e logiche di funzionamento:

- $\overline{\text{BUSY}}$  (pin 11) vale 128 se è al livello basso (logica negativa).
- ACK (pin 10) vale 64 se è al livello alto (logica positiva).
- PAPER OUT (pin 12) vale 32 se è al livello alto (logica positiva).
- SELECTED (pin 13) vale 16 se è al livello alto (logica positiva).
- ERROR (pin 15) vale 8 se è al livello alto (logica positiva).

LINEE	BUSY	ACK	PAPER OUT	SELECTED	ERROR	1	1	1
PIN	11	10	12	13	15			

I rimanenti 3 bit meno significativi del byte di ingresso non sono accessibili sul connettore e sono poste, internamente all'interfaccia, al livello alto (1112 = 710 ).

Avendo a disposizione 5 bit (32 combinazioni) è possibile acquisire, via software in una variabile A, un valore compreso tra 0 e 31.

Ciò si realizza con i seguenti passaggi:

**A = A XOR 128** (per complementare il bit applicato su BUSY);

**A = A - 7** (per azzerare i tre bit meno significativi : operazione, comunque, non necessaria);

**A = A / 8** (per lo scorrimento a destra di 3 posizioni).

L'algoritmo di acquisizione in linguaggio QBASIC risulta, pertanto:

**A = ((INP(889) XOR 128) - 7 ) / 8**

In questo modo la parola di ingresso A assume un valore compreso tra 0 e 31.

### Registro di controllo (indirizzo 890):

L'indirizzo Centronics 890 (37A esadecimale) rende disponibile in uscita altri 4 bit di cui tre attivi in logica negativa ed una in logica positiva :

$\overline{\text{STB}} = \overline{\text{STROBE}}$  (pin 1) vale 1 se è al livello basso (logica negativa).

$\overline{\text{AF}} = \overline{\text{AUTOFEED}}$  (pin 14) vale 2 se è al livello basso (logica negativa).

IP = INIZIALIZE PRINTER (pin 16) vale 4 se è al livello alto (logica positiva).

$\overline{\text{SI}} = \overline{\text{SELECT INPUT}}$  (pin 17) vale 8 se è al livello basso (logica negativa).

IRQE = ABILITA INTERRUPT vale 16 se è al livello alto (logica positiva)  
ma non è disponibile fisicamente.

Indirizzo 890	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LINEE				IRQE	SI	IP	AF	STB
PIN					17	16	14	1

Se si intende trasmettere una parola a 4 bit con valore compreso tra 0 e 15 e con i bit attivi in logica positiva, occorre commutare (XOR con 1 logico) i bit **SI e AF,STB** e lasciare inalterato (XOR con zero logico) il bit IP

### OUT 890, A XOR 11

A3	A2	A1	A0	A nel Pc
1	0	1	1	11
A3	A2	A1	A0	A XOR 11
A3	A2	A1	A0	A in uscita

### **Procedura per dichiarare di ingresso la porta all'indirizzo 888.**

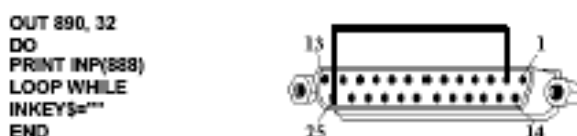
Se l'interfaccia Centronics è impostata come EPP (Enhanced Parallel Port) è possibile effettuare l'input di un byte all'indirizzo 888 purché si imposti ad 1 il bit 5 del registro di controllo posto all'indirizzo 890.

Impostando a 0 il bit 5 dell'indirizzo 890, il byte DATA8...DATA1 del registro dati all'indirizzo 888 ritorna a funzionare come byte di output.

Si riporta un semplice listato di test in QBASIC (facilmente modificabile in un altro linguaggio di programmazione) per provare la Centronics all'indirizzo 888 come canale di input a 8 bit.

Con un semplice spezzone di cavo telefonico (quello usato per collegamenti su bread-board) si collega il pin 2 (DATA1) al pin 25 (GND) del connettore (femmina) a 25 poli posto sul retro del PC e si prova il programma ottenendo sul monitor la stampa continua del numero 254 fino alla pressione di un tasto qualsiasi sulla tastiera. Portando il filo sul pin 3, 4, ecc. tenendo l'altro capo del filo sempre a massa, si può verificare che effettivamente il dato in lettura si modifica in funzione del pin collegato a massa. È superfluo dire che le linee DATA8 ... DATA1 posti sui pin 9...2 sono da ritenersi allo stato 1 se tenute aperte.

Per impostare l'interfaccia Centronics come EPP, infine, si va nel BIOS SETUP e si imposta la stampante come EPP anziché come SPP.

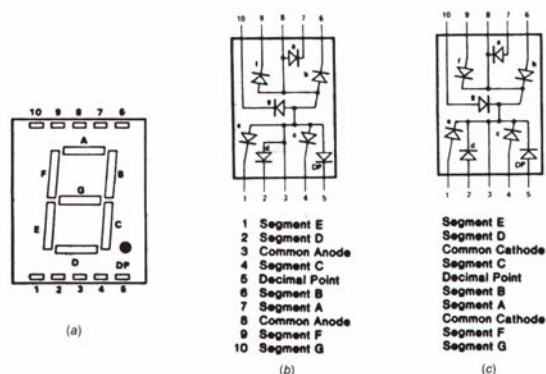


**Figura 6 - Programma e collegamenti per provare la Centronics come porta di ingresso all'indirizzo.**

### **Display LED a sette segmenti**

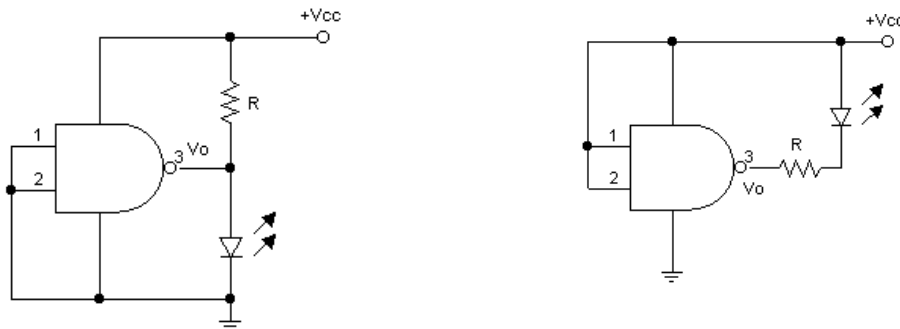
È uno tra i display più usati: è realizzato a 7 segmenti ognuno dei quali è costituito da uno o più LED collegati in modo da ottenere la classica disposizione di figura 11a. Il dispositivo con struttura dual-in-line, possiede 7 ingressi denominati *a,b,c,d,e,f,g*, per ogni segmento più un ingresso per il punto decimale DP (Decimal Point).

Un display è detto a catodo comune se i 7 catodi dei LED sono connessi insieme e gli anodi sono disponibili all'utente come in figura 7b. Se invece gli anodi sono connessi insieme il display è detto a anodo comune, come mostrato in figura 7c.



**Figura 7a) – Display a sette segmenti; b) disposizione ad anodo comune; c) disposizione a catodo comune.**

Per illuminare un segmento è necessario che il corrispondente LED sia attraversato da corrente, cioè l'anodo deve risultare positivo rispetto al catodo; inoltre per limitare la corrente in ogni LED la valore tipico di circa 10 - 20 mA, si deve inserire una resistenza di limitazione come mostrato in figura 8 nel caso si consideri uno solo dei sette LED.



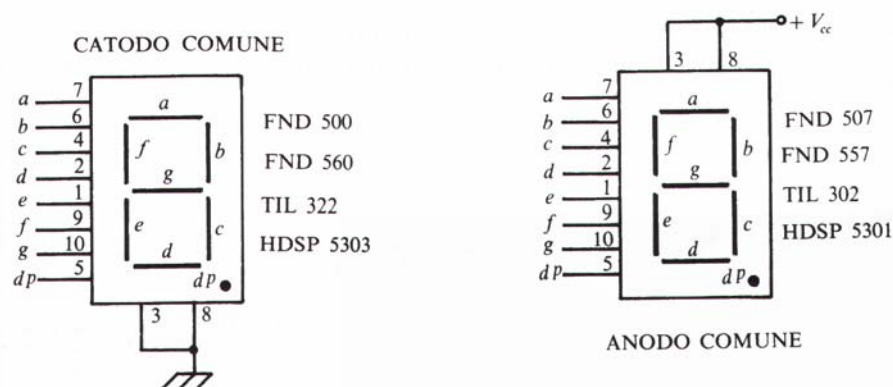
**Figura 8 – Circuiti per l'accensione di un diodo LED; a) collegamento a catodo comune; b) collegamento ad anodo comune; Nel primo caso il LED è acceso se l'uscita della porta è al livello alto, viceversa nel secondo caso.**

Se , ad esempio, si utilizza un diodo LED con le seguenti caratteristiche:

$V_D = 1.8 V$  e  $I_D = 10 mA$  si ha:

$$R = \frac{V_{CC} - V_D}{I_D} = \frac{5 - 1.8}{10 \cdot 10^{-3}} = 320 \Omega$$

In figura 9 si mostrano le piedinature di alcuni display ad anodo ed a catodo comune. Dalla 13 a si deduce che l'accensione di un segmento di un display a catodo comune si ottiene inviando un livello logico alto all'anodo, mentre nel display ad anodo comune si deve porre un livello logico basso sul catodo.



**Figura 9 – Display LED a sette segmenti.**

## Elenco componenti

Si riporta l'elenco dei componenti utilizzati per la realizzazione del progetto e la relativa strumentazione.

- 1 Display LED a 7 seg. (catodo comune) TDSR 5160 H
- 74LS244
- Resistenze 7 x 150  $\Omega$
- Resistenze 8 x 10 K $\Omega$
- 8 Interruttori
- 1 Connettore Centronics

Alimentatore + 5 Vcc, Multimetro, PC.

## Risoluzione

Sia DIN7 l'ingresso a maggior priorità e DIN0 quello a priorità più bassa. Si suppone di voler lavorare in logica positiva: un livello logico alto applicato ad una linea di ingresso viene riconosciuto come ingresso abilitato. Si consiglia, infine, di inserire 8 resistenze di pull-up, ciascuna da 10 K $\Omega$ , tra l'alimentazione + 5V e ciascun ingresso DIN7...DIN0. Anche le linee di uscita lavorano in logica positiva, poiché si è detto di utilizzare un display a catodo comune. Si riportano i possibili stati delle linee di ingresso, i codici numerici acquisiti e quelli di uscita che consentono il pilotaggio del display.

DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0	DIN	N1	g	f	e	d	c	b	a	N2
1	X	X	X	X	X	X	X	>127	7	0	0	0	0	1	1	1	7
0	1	X	X	X	X	X	X	>63	6	1	1	1	1	1	0	1	125
0	0	1	X	X	X	X	X	>31	5	1	1	0	1	1	0	1	109
0	0	0	1	X	X	X	X	>15	4	1	1	0	0	1	1	0	102
0	0	0	0	1	X	X	X	>7	3	1	0	0	1	1	1	1	79
0	0	0	0	0	1	X	X	>3	2	1	0	1	1	0	1	1	91
0	0	0	0	0	0	1	X	>1	1	0	0	0	0	1	1	0	3
0	0	0	0	0	0	0	1	>0	0	0	1	1	1	1	1	1	63
0	0	0	0	0	0	0	0	>0	8	0	0	0	0	0	0	0	0

Il software, dopo aver acquisito il dato DIN, deve determinare in numero N1 secondo il seguente ragionamento. Se  $DIN > 127$  ( $2^7 - 1$ ) allora sicuramente  $DIN7 = 1$  e, indipendentemente dallo stato degli altri ingressi, si deve porre  $N1=7$  poiché  $DIN7$  è l'ingresso a maggior priorità. In caso negativo si deve controllare se  $DIN > 63$  ( $2^6 - 1$ ). In caso affermativo si ha  $N1=6$  altrimenti si procede controllando  $DIN$  con 15,7, ecc.

E' sufficiente attivare un ciclo definendo una variabile contatore I da 7 a 0 ed imporre i controlli visti prima fermandosi quando questo è soddisfatto. In tal caso  $N1$  è uguale al valore corrente di I. se il controllo non è mai soddisfatto, vuol dire che non è attivato alcun ingresso per cui  $DIN=0$  e per  $N1$  attribuiremo convenzionalmente il valore 8 (display spento). Il valore  $N2$  da fornire in uscita per il pilotaggio del display è uno dei 9 valori memorizzati in un vettore.

In figura 10 vi è mostrato lo schema elettrico del circuito modificato, in quanto sono state aggiunte le 8 resistenze di pull-up ciascuna da 10 K $\Omega$ , tra l'alimentazione e ciascun ingresso DIN7...DIN0. Inoltre sono state inserite le resistenze di protezione per il display ciascuna per ogni segmento da 150  $\Omega$ . Infine gli otto microswitch sono stati sostituiti con otto interruttori meccanici di tipo on-off.



Si riporta il listato del programma in Visual Basic 6 che gestisce il codificatore.

Il programma consente la visualizzazione del formato binario e dell'ingresso attivato con maggior priorità anche inserendo manualmente un valore nella casella di testo text3, etichettata DIN.

```
Dim n2(10) As Integer
Private Sub Form_Load()
n2(7) = 7
n2(6) = 125
n2(5) = 109
n2(4) = 102
n2(3) = 79
n2(2) = 91
n2(1) = 6
n2(0) = 63
n2(8) = 0
End Sub

Private Sub Text3_Change()
visualizzazione
End Sub

Private Sub cmdCodifica_Click()
aquisizione
visualizzazione
End Sub

Private Sub cmdEsci_Click()
Dim esci As Integer
esci = MsgBox("Sei Sicuro?", vbInformation + vbOKCancel, "Uscita")
If esci = vbOK Then End
End Sub

Private Sub aquisizione()
Out 890, 1
ah = (Inp(889) And 120) * 2
Out 890, 2
al = (Inp(889) And 120) / 8
a = ah + al
Out 890, 0
Text3.Text = a
End Sub

Private Sub visualizzazione()
din = Val(Text3.Text)
If din > 255 Then
    din = 0
    Text3.Text = 0
End If
n1 = 8
For i = 0 To 7
    If din > 2 ^ i - 1 Then n1 = i
Next i
Text2 = ""
If Not n1 = 8 Then Text2 = n1
Out 888, n2(n1)
For i = 0 To 7
    Text1(i) = din Mod 2
    din = Int(din / 2)
Next i
End Sub
```

La routine **Form\_Load()** si attiva al caricamento del programma e consiste nell'assegnare al vettore n2 i relativi 9 valori come si è visto nella tabella 14.

La routine **Text3\_Change()** si attiva quando il contenuto della casella di testo denominata DIN cambia il valore. Il suo compito è quello di attivare la routine *visualizzazione* che consente di materializzare il numero decimale DIN in forma binaria e nella casella di testo a destar compare il numero di posizione del bit più significativo posto ad 1.

La routine **CmdCodif\_Click()** si attiva facendo click sul pulsante *Codifica*. Il suo compito +è quello di eseguire l'acquisizione di un byte dagli otto interruttori e la visualizzazione su monitor e sul display.

La routine **CmdEsci\_Click()** si attiva facendo click sul pulsante *Esci*. Il suo compito è quello di interrompere l'esecuzione del programma chiedendone la conferma.

La routine **acquisizione** cattura il byte prodotto tramite interruttore in due fasi da 4 bit. In ciascuna di tali fasi, nel registro di stato devono essere presenti solo i 4 bit acquisiti che occupano le posizioni 6.5.4. e 3 del registro. Per essere sicuri che gli altri bit del registro non influenzino l'acquisizione, si procede all'azzeramento del bit 7 e dei bit 2.1 e 0. Questo azzeramento si ottiene eseguendo il prodotto logico, bit a bit, tra il contenuto del registro e il numero corrispondente alla configurazione binaria: 01111000 che corrisponde al decimale 120. Infatti:

Registro di stato	A7	A6	A5	A4	A3	A2	A1	A0
Byte di maschera	0	1	1	1	1	0	0	0
Risultato prodotto logico	0	A6	A5	A4	A3	0	0	0

La moltiplicazione per due consente lo scorrimento a sinistra di una posizione. Il risultato è posto nella variabile denominata AH. La divisione per  $8=2^3$  consente lo scorrimento a destra di tre posizioni. Il risultato è posto nella variabile denominata AL. La somma tra AH e AL è posta nella variabile A che viene poi inviata nella casella di testo DIN.

AH	DIN7	DIN6	DIN5	DIN4	0	0	0	0
AL	0	0	0	0	DIN3	DIN2	DIN1	DIN0
A	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0

L'istruzione OUT 890, 1 porta al livello basso  $\overline{STB}$  che abilita i primi 4 buffer tri-state a cui sono collegate le linee DIN7 Din6 DIN5 DIN4.

L'istruzione OUT 890, 2 porta al livello basso  $\overline{AF}$  che abilita gli ultimi 4 buffer tri-state a cui sono collegate le linee DIN3 DIN2 DIN1 DIN0.

La routine **visualizzazione** individua l'ingresso abilitato avente maggior priorità da visualizzare nella casella di testo text2, la più grande, e poi converte il numero acquisito in binario a visualizzare su monitor.

## Istruzioni di Input e Output in Visual Basic 6

Il linguaggio di programmazione Visual Basic consente di mettere a punto interfacce grafiche estremamente elaborate ed accattivanti con poca fatica, ma purtroppo, in forma nativa, **non sopporta le istruzioni di input e di output** direttamente da unità periferiche. Per ovviare a ciò è possibile utilizzare un particolare file di libreria DLL, come ad esempio **INPOUT32.DLL**, funzionante su computer Windows 95,98,ME ma non con Windows 2000.

Questo file dovrà essere copiato nella cartella C:\Windows\System del proprio hard-disk. In questo modo è possibile utilizzare in Visual Basic 4 e versioni successive le istruzioni OUT e INP. Affinché si possa invocare nel nostro applicativo, il file di libreria **INPOUT32.DLL** si deve includere nel progetto Visual Basic da noi generato il modulo **INPOUT32.BAS**.

Il file libreria è di seguito riportato:

```
'Inp and Out declarations for direct port I/O
'in 32-bit Visual Basic 4 programs.
```

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Nel caso in cui si opera in ambiente Windows 200, 200XP o NT è necessario scaricare il file *Windows 95NT Port I/O driver* che consente di caricare il file *port95nt.exe* che installa i necessari componenti. Il file è disponibile gratuitamente al sito [www.driverlinx.com](http://www.driverlinx.com)

## Realizzazione pratica

La realizzazione pratica del progetto si è sviluppata in diverse fasi di lavorazione. La prima operazione è quella di montare il circuito illustrato in figura 10 su brad-board per verificarne il funzionamento. Collaudata questa prima fase, si passa alla realizzazione del circuito stampato monofaccia che comprende:

- **LAYOUT E SBROGLIO.** Prima di arrivare al master vi sono due fasi fondamentali: lo studio del layout e lo sbroglio. Il layout è la disposizione dei componenti sulla scheda, mentre lo sbroglio consiste nel definire il miglior percorso delle piste che realizza tutti i collegamenti elettrici previsti, senza incroci di poligoni. Il prodotto finale, lo sbrogliato, costituisce la bozza del tracciato, che dopo successivi perfezionamenti darà luogo al master. Si noti che per ottenere lo sbrogliato si è utilizzato il PC con un programma specifico nel realizzare circuiti stampati;
- **MASTER.** Su un nuovo foglio di raster si trasferisce lo sbrogliato precedentemente eseguito, prestando molta attenzione all'esattezza delle dimensioni meccaniche;
- **ESECUZIONE DEL MASTER.** Sulla superficie ramata della basetta di bachelite presensibilizzata, di dimensioni 10x12, si è riportato il master del circuito, (naturalmente relativo al lato piste), con la tecnica della foto incisione;
- **FOTOINCISIONE.** Per asportare dalla basetta le parti di rame non ricoperte dal mezzo protettivo, in maniera da lasciare soltanto il tracciato del circuito stampato. Su di un foglio lucido si è riportato esattamente il master del circuito. Nel bromografo si è lasciata la basetta presensibilizzata con sopra il foglio lucido con le piste riportate in figura 18. Dopo circa 10 minuti si toglie la basetta dal bromografo è senza far prendere molta luce si passa nella soda caustica per lo sviluppo del basetta, esattamente con lo sviluppo delle fotografie. Infine si immerge la basetta nel per cloruro ferrino per circa 1 ora è rimangono le piste di collegamento del circuito;
- **REALIZZAZIONE DEI FORI.** Poiché è scelto di realizzare la basetta mediante la tecnica THT (Through Hole Technology), ossia la tecnica a foro passante, si è proceduto alla foratura della basetta nei punti relativi all'inserzione dei reofori dei vari componenti. Si noti che l'operazione di foratura è stata preceduta da un'operazione di bulinatura;

- **FISSAGGIO DEI COMPONENTI.** Si è passati, successivamente al fissaggio dei componenti sulla basetta. I reofori, fatti sporgere dal lato piste, sono stati dunque saldati determinando l'effettiva connessione elettrica tra i rispettivi componenti.

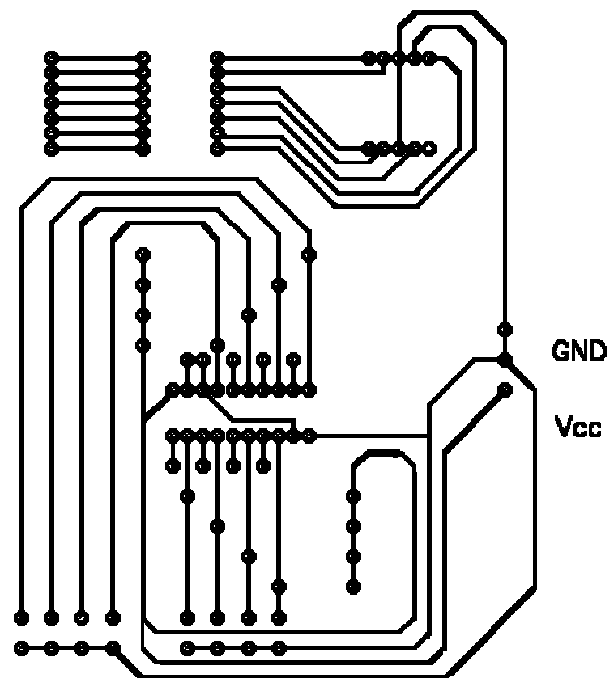


Figura 12 – Master del circuito relativo al lato piste.

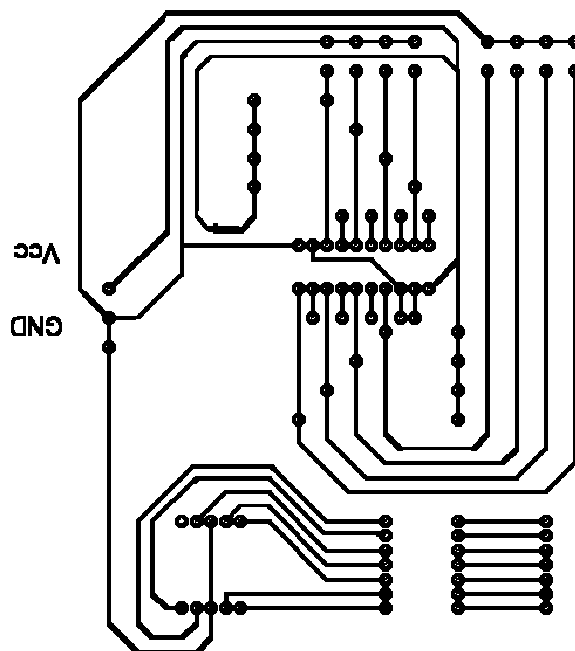


Figura 14 – Master del circuito relativo al lato componenti.

Infine si è preparata la scatola che contiene lo stampato, realizzando i relativi fori e incassi per gli interruttori, connettore Centronics e display LED a sette segmenti.  
In figura 15 si mostra il risultato finale.



**Fig. 15**