

*Istituto Tecnico Industriale Statale PANETTI – BARI  
AREA DI PROGETTO DELLA 5 ETB  
CORSO ELETTRONICA E TELECOMUNICAZIONI  
ANNO SCOLASTICO 2003-2004*

*Coordinatore:*

Prof. Ettore Panella

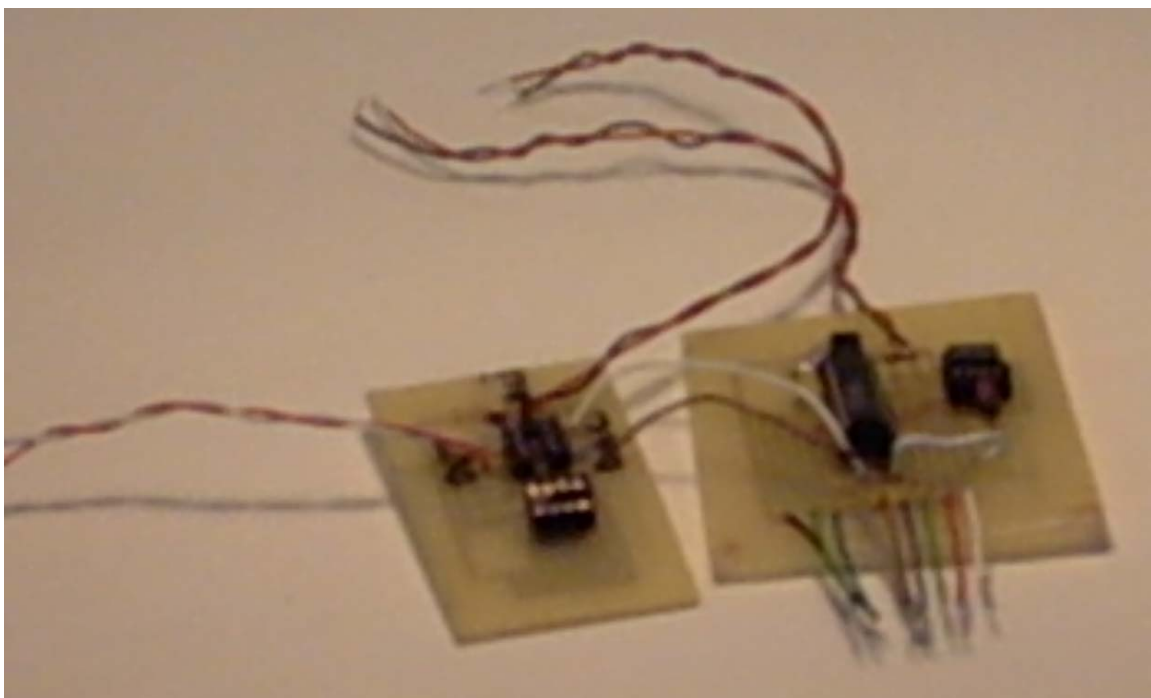
[www.ettorepanella.com](http://www.ettorepanella.com)

*Studente:*

Pachito Marco Calabrese

[calabrese1957@libero.it](mailto:calabrese1957@libero.it)

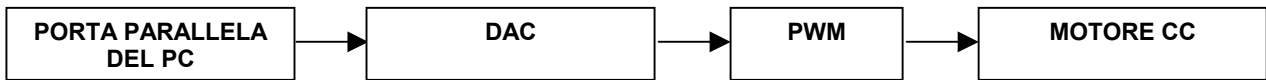
**PILOTAGGIO DI UN MOTORE IN CORRENTE CONTINUA CON  
TECNICA PWM COMANDATO CON LA PORTA PARALLELA DI UN PC  
E SOFTWARE DI GESTIONE IN VISUAL BASIC 6.**



## GENERALITA'

L'obiettivo del lavoro di classe è la progettazione, realizzazione e collaudo di un sistema in grado di controllare la velocità di un motore in corrente continua mediante l'ausilio di un PC.

In fig. 1 si mostra lo schema a blocchi del sistema.



**Fig. 1 – Schema a blocchi del sistema di controllo della velocità di un motore in cc.**

Si descrive brevemente il principio di funzionamento del sistema.

Il software in VB6 genera numeri binari a 8 bit sulla porta Centronics del PC.

Il convertitore DAC 0830 della National è stato configurato in modo da produrre una tensione di uscita variabile da 0-10V quando il numero binario di entrata varia tra 0-255 in decimale.

La tensione di uscita del DAC indicata con  $V_o$  comanda un circuito PWM (Pulse Width Modulation) in grado di generare un treno di onde rettangolari a frequenza fissa e Duty Cycle variabile con  $V_o$ . Il segnale di uscita del circuito PWM rappresenta la tensione di comando del motore. La teoria della tecnica PWM dimostra che la potenza di comando del motore, e quindi la sua velocità, sono direttamente proporzionali al valore del Duty Cycle.

In definitiva se tramite software si varia il codice numerico si varia il Duty Cycle e quindi la velocità del motore.

Si descrivono i blocchi del sistema:

- Motore in corrente continua
- Pwm
- Dac
- Porta Centronics
- Software in VB6
- Realizzazione circuito stampato

## MOTORE IN CORRENTE CONTINUA

Il motore in corrente continua è un dispositivo, o meglio attuatore, in grado di trasformare una tensione continua di entrata in una coppia motrice di uscita, quindi energia elettrica in meccanica.



**Fig. 2 – Motorino elettrico di bassa potenza**

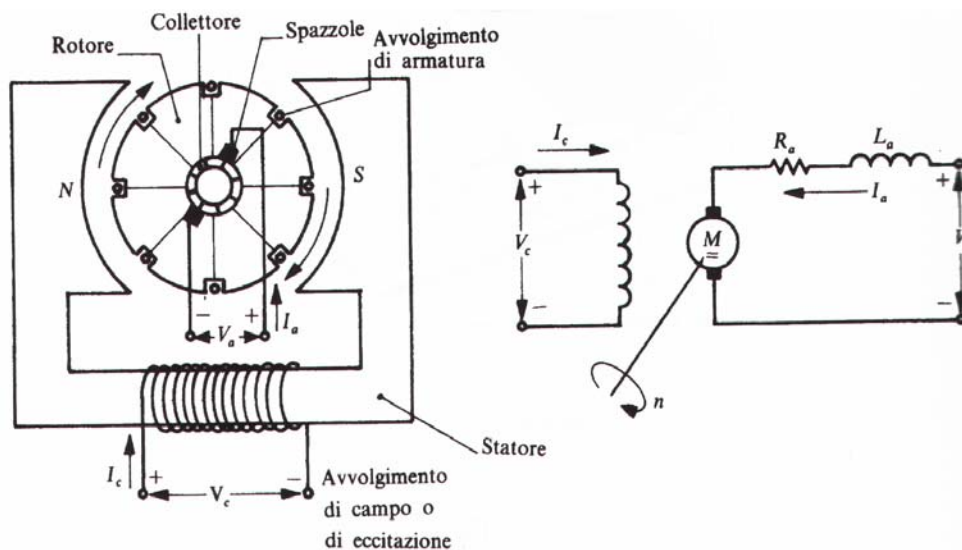
In figura 3 è mostrata la schematizzazione costruttiva di un motore in corrente continua ad eccitazione indipendente.

Il motore in cc. è costituito da uno *statore* (la parte statica) e da un *rotore* (la parte mobile). Lo statore genera un flusso magnetico il cui valore dipende dall'intensità di corrente  $I_c$  del circuito di eccitazione. Il rotore è posto entro le espansioni polari dello statore. Esso è costituito di materiale ferroso o plastico e presenta sulla periferia delle cave entro le quali è posto un avvolgimento da cui vengono derivati dei conduttori che sono connessi, in modo opportuno, alle lamelle di un dispositivo denominato *collettore*.

Sul collettore poggiano dei contatti striscianti (*spazzole*) generalmente di carbone attraverso cui si alimenta il motore. Il rotore, le spazzole e il collettore costituiscono il *circuito di armatura* del motore.

Con  $V_a$  e  $I_a$  si indicano rispettivamente la tensione e la corrente di armatura o alimentazione del motore del motore.

Lo statore insieme all'avvolgimento di eccitazione, costituisce il *circuito di campo* cui applicare la tensione  $V_c$  e la corrente  $I_c$  che definiscono il campo magnetico induttore. Sono possibile altri due modi di eccitazione.



**Fig. 3 – Schematizzazione di un motore in cc**

In un motore in cc il numero di giri/min  $n$  dipende dalla tensione applicata  $V_a$  e dal flusso magnetico induttore  $\Phi$  secondo la relazione:

$$n \cong \frac{V_a}{K \cdot \Phi};$$

dove  $K$  è una costante costitutiva del motore  $V_a$ . La precedente relazione vanta la diretta proporzionalità tra  $n$  e  $V_a$ . La coppia motrice  $C_m$  sviluppata dal motore vale:

$$C_m = K_m \cdot \Phi \cdot I_a$$

Dove  $K_m = \frac{60 \cdot K}{2\pi}$  e  $I_a$  è la corrente di armatura.

Si riporti i dati di un piccolo motore in cc:

1. tensione nominale:  $V_a = 32 \text{ V}$ ;
2. corrente assorbita a rotore bloccato:  $I_{a\max} = 4 \text{ A}$ ;
3. velocità a vuoto:  $n = 1650 \text{ giri/min}$ ;
4. potenza nominale:  $P = 30 \text{ W}$  ;
5. coppia allo spunto:  $C_m = 0.7 \text{ Nm}$ ;
6. resistenza di armatura:  $R_a = 8 \Omega$
7. induttanza di armatura:  $L_a = 13.68$
8. costante di tempo elettrica:  $T_e = L_a / R_a = 1.7 \text{ msec}$
9. momento di inerzia:  $J = 0.05 \text{ Kgm}^2$ ;
10. costante di tempo meccanica:  $\tau_m = \frac{J \cdot R_a}{K \cdot K_m} = 10.6$

## TECNICA PWM

In fig. 4 si riporta lo schema elettrico del blocco PWM, impiegato, formato da 2 timer 555. Il primo in configurazione in configurazione astabile, il secondo in configurazione monostabile. La durata dell'ingresso generato del monostabile dipende dall'ampiezza della tensione proveniente dal DAC.

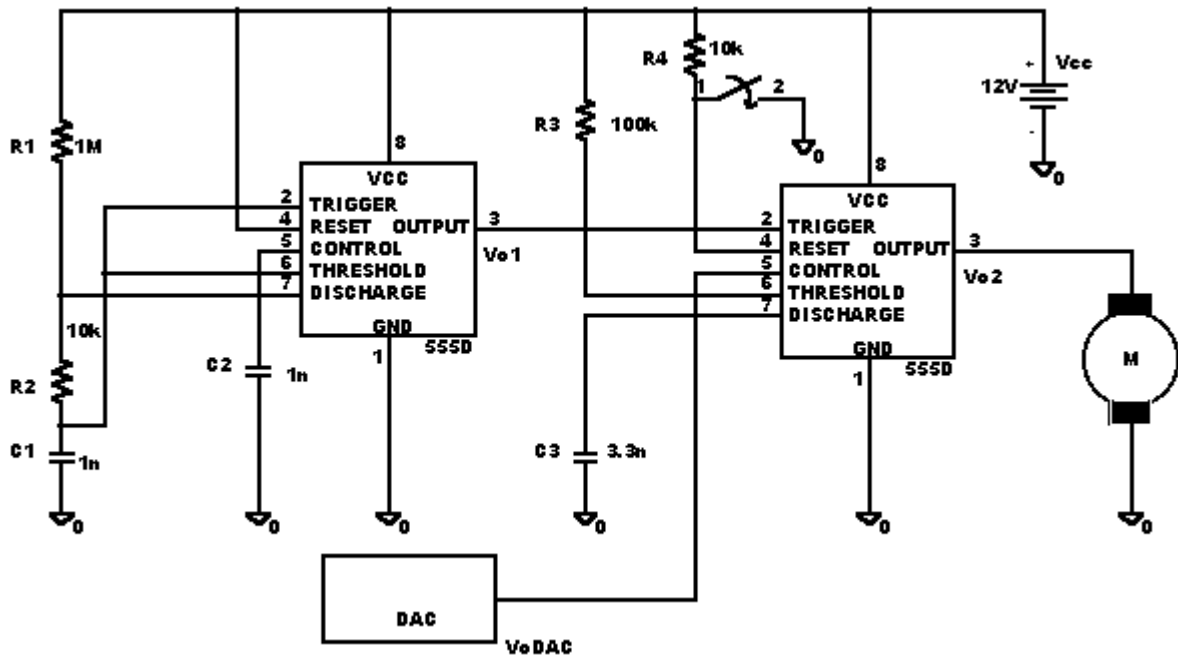


Fig. 4 – Schema del PWM controllato in tensione

Il circuito PWM consente di generare un segnale impulsivo a frequenza fissa e a Duty-Cycle variabile.

Il primo stadio è in configurazione astabile, sul pin 3 viene generata un'onda impulsiva avente le seguenti caratteristiche.

$$T_1 = 0,693 \cdot (R_1 + R_2) \cdot C \cong 700 \mu s$$

$$T_2 = 0,693 CR_2 \cong 7 \mu s$$

quindi si ha:  $T = T_1 + T_2 = 707 \mu s$

$$f = \frac{1}{T} \cong 1,41 \text{ K Hz}$$

Per un corretto funzionamento la frequenza dell'astabile nel PWM deve essere sufficientemente superiore alla frequenza di taglio  $f_t$  del motore. Normalmente si pone  $f > 10f_t$ . Per un motore in corrente continua si ha:

$$f_t = \frac{1}{2\pi(\tau_e + \tau_m)} \cong 100 \text{ Hz}$$

Dove  $\tau_e$  e  $\tau_m$  sono, rispettivamente, la costante di tempo elettrica e la costante di tempo meccanica del motore. Normalmente  $\tau_m > \tau_e$ . Tipicamente  $\tau_m$  è dell'ordine della decina di millisecondi. Le precedenti considerazioni giustificano la scelta della frequenza dell'astabile.

Il secondo stadio è montato da monostabile controllato dalla tensione  $V_{O\text{dac}}$  (proveniente dal DAC) variabile da 0÷10 V.

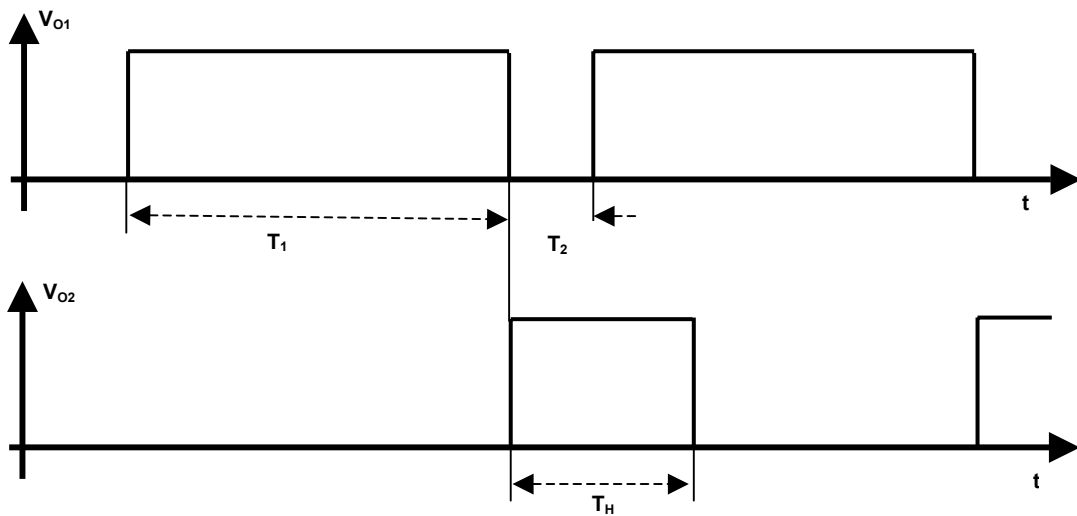
Si dimostra che il monostabile è sensibile ai fronti di discesa del segnale  $V_{O1}$  applicato al pin 2 del secondo Timer 555.

L'impulso di uscita  $V_{O2}$  ha una durata:

$$T_H = R_3 \cdot C_3 \cdot \ln \left( \frac{V_{cc}}{V_{cc} - V_i} \right)$$

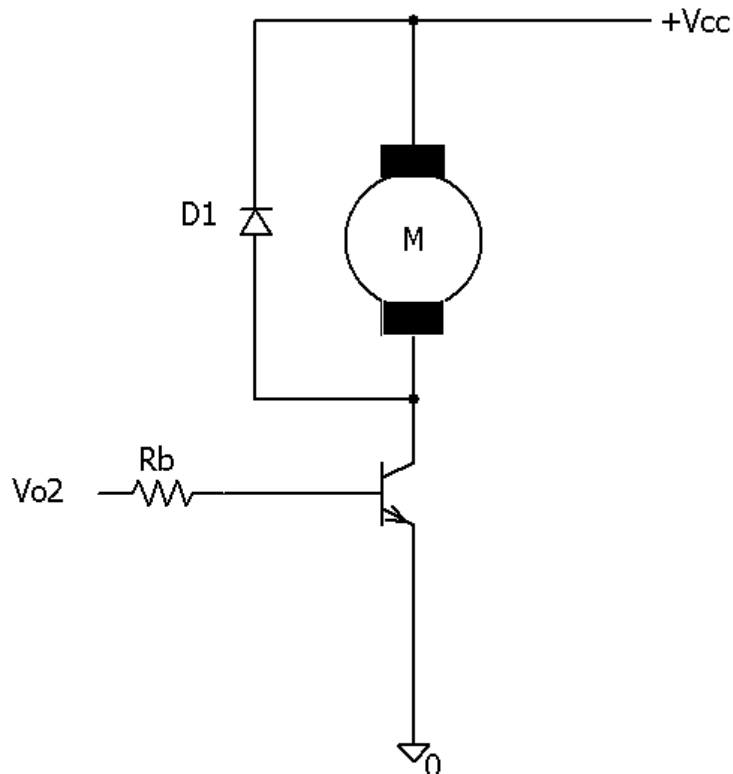
Si evince che al variare di  $V_{O\text{dac}}$  varia  $T_H$ .

In fig.5 si riportano le tempificazioni dei segnali di  $V_{O1}$  e  $V_{O2}$ .



**Fig. 5 - Diagramma di tempificazione**

Il segnale  $V_{O2}$ , avendo scelto un motore di piccola potenza, pilota direttamente il motore. In caso contrario è opportuno porre tra il monostabile e motore un dispositivo di potenza, vedi fig.6, come ad esempio un transistor BJT.



**Fig. 6 - Circuito di potenza con transistor BJT**

Tale segnale è caratterizzato da un periodo costante  $T=707\mu s$  coincidente con quello di  $V_{O2}$ , e un impulso di livello alto variabile con  $V_i$ .

Per un corretto funzionamento del PWM,  $T_H$  deve essere compreso tra 0 e  $T$ .

Nel nostro caso per:

$$V_i = 0 \text{ V si ha } T_{Hmin} = 0$$

invece per

$$V_i = 10 \text{ V e } V_{cc} = 12 \text{ V}$$

si ha

$$T_H = 10^{-5} \cdot 3,9 \cdot 10^{-9} \ln 6 = 690 \mu s$$

$$D = \frac{T_H}{T} = \frac{690 \cdot 10^{-6}}{707 \cdot 10^{-6}} \cong 0,97$$

e quindi il Duty-Cycle varia tra circa il 0% e il 97%

Il motore è sotto posto al segnale impulsivo a Duty-Cycle variabile proveniente da  $V_{O2}$ .

Il motore è comandato da un segnale con valore efficace pari a:

$$V_{\text{eff}} = \sqrt{\frac{1}{T} \cdot \int_0^T V_{O2}^2 \cdot dt} =$$

$$V_{\text{eff}} = \sqrt{\frac{1}{T} \cdot V_{CC}^2 \cdot dt} =$$

$$V_{\text{eff}} = \sqrt{\frac{T_H}{T}} =$$

$$V_{\text{eff}} = \sqrt{D} =$$

Quindi il motore si può pensare alimentato a una tensione continua di valore  $V$  proporzionale a  $\sqrt{D}$ . Indicando  $R_M$  la

resistenza del motore si ricava che la potenza sviluppata dal motore vale  $P = \frac{V^2}{R_M} = \frac{V_{CC}^2}{R_M} \cdot D$ .

Indicando con  $P_{\text{max}} = \frac{V_{CC}^2}{R_M}$  si ha  $P = P_{\text{max}} \cdot D$ .

$P_{\text{max}}$  rappresenta la potenza sviluppata dal motore se fosse alimentato continuamente dalla  $V_{CC}$ .

$P$  rappresenta la potenza del motore sotto controllo PWM:

Per  $D=0$   $P=0$  e il motore è fermo, invece, per  $D=1$  il motore è alla massima velocità.

Nel nostro caso si otterrà una potenza di lavoro pari al 97% di quella massima.

## TIMER 555

Il timer 555 è un circuito integrato a 8 piedini progettato specificatamente per funzionare da multivibratore stabile e monostabile. Deve il suo nome al fatto di possedere al suo interno tre resistenze ciascuna di valore 5 k $\Omega$ . Nonostante la Signetics lo abbia lanciato sul mercato dal 1972, il timer 555 viene ancora largamente utilizzato grazie alla sua versatilità. Fra le caratteristiche più importanti ricordiamo:

- Singola alimentazione regolabile tra 5 V e 15 V;
- Durata dell'impulso o periodo di oscillazione regolabile da alcune decine di microsecondi ad alcune decine di minuti;
- Capacità di carico fino a 200 mA;
- Uscita CMOS o TTL compatibile se si utilizza la stessa  $V_{CC}$ ;
- Stabilità in temperatura migliore dello 0.005% per grado centigrado.

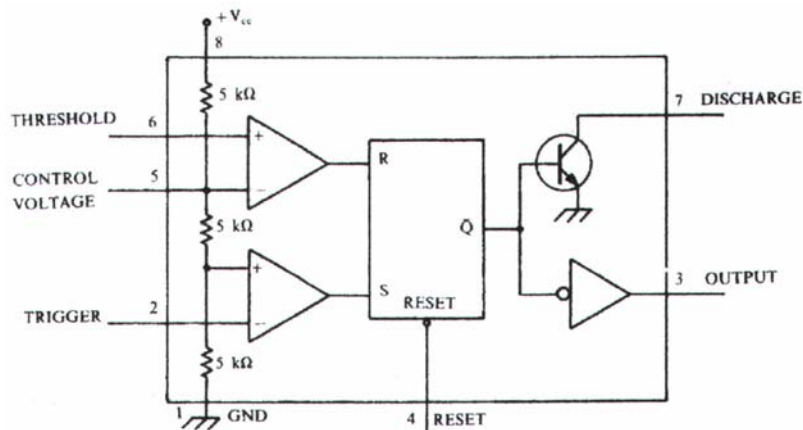


Fig. 7 - Schema interno del Timer 555

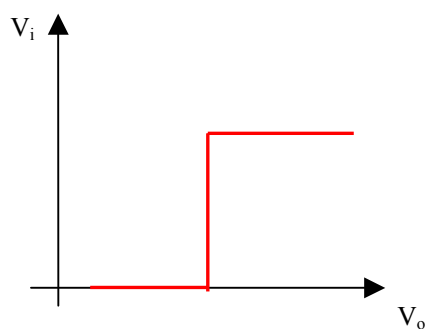
Il timer 555 risulta essere costituito da due comparatori (invertente e non), un flip-flop S-R, un BJT NPN, un buffer invertente, da tre resistenze in serie da  $5\text{ k}\Omega$ . Il piedino 1, indicato con GND è la massa mentre il piedino 8, indicato con  $V_{cc}$  è quello a cui va applicata la tensione di alimentazione. Il piedino 6 *threshold* è collegato all'ingresso non invertente del comparatore che pilota l'ingresso di reset del flip-flop mentre l'ingresso invertente dello stesso comparatore è polarizzato

al valore di  $2\frac{V_{cc}}{3}$  tramite il partitore resistivo se si lascia flottante l'ingresso *control voltage* posto al piedino 5 (in tal caso

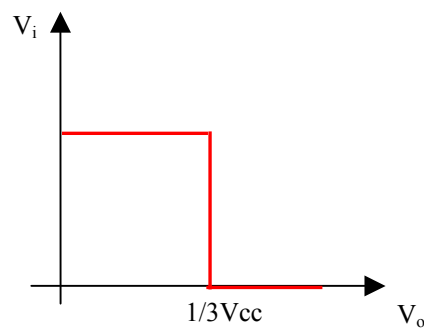
si preferisce collegare un condensatore tra il piedino 5 e massa che filtri gli eventuali disturbi di alta frequenza che altererebbero i riferimenti dei comparatori). Il piedino 2, detto *trigger*, è collegato all'ingresso invertente del comparatore che pilota l'ingresso di set del flip-flop mentre l'ingresso non invertente dello stesso comparatore è polarizzato al valore

$\frac{V_{cc}}{3}$  dal precedente partitore. Applicando una tensione esterna all'ingresso 5 direttamente o tramite un'opportuna

resistenza, si modifica il rapporto di partizione delle tensioni applicate ai comparatori. Il piedino 4 è collegato all'ingresso di azzeramento del flip-flop e funziona in logica negativa. Il piedino 7, indicato con *discharge*, è il collettore aperto del BJT la cui base è connessa all'uscita complementata del flip-flop e l'emettitore è collegato a massa. Il piedino, indicato con *output*, è l'uscita del buffer invertente anch'esso pilotato dall'uscita complementata dal flip-flop.



Comparatore non invertente

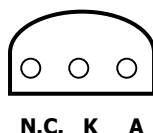


Comparatore invertente

### LM336

L' LM336 è un diodo Zener di precisione con  $V_{ref} = 2,5\text{ V}$  composto da tre pin A anodo K catodo e N.C. non connesso.

Vista dall'alto:



N.C. K A

## DAC

Il DAC (Digital to Analog Converter) è un dispositivo in grado di convertire un numero binario a  $n$  bit in un valore di tensione o di corrente;  $n$  indica la risoluzione del convertitore, ovvero un convertitore DAC a 16 bit, ad esempio, consente di ottenere una uscita definibile in 65536 livelli.

Il livello massimo del segnale viene definito attraverso un tensione di riferimento che può essere esterna al DAC (definibile dal progettista) oppure interna (tensione fissa, spesso coincidente con la tensione di alimentazione).

L'uscita è tipicamente lineare, ma potrebbe essere anche esponenziale al fine di ottenere una maggiore dinamica come richiesto in applicazioni di trasmissione audio.

Il dato di ingresso al DAC può essere fornito in forma parallela ad  $n$  ingressi oppure in forma seriale. In questo caso il dato binario di ingresso viene fornito da un dispositivo a microcontrollore attraverso un opportuno bus seriale.

### DAC 0830

Il DAC 0830, utilizzato nel progetto, è un Convertitore Digitale Analogico della National che utilizza la tecnica della rete a scala R-2R (resistor ladder network).

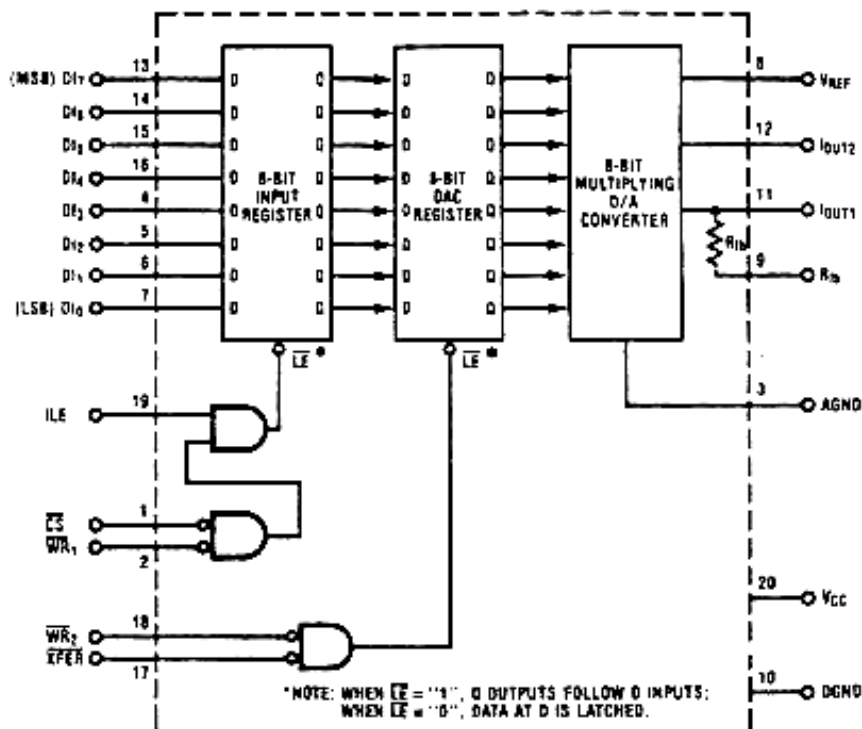


Fig. 8 – Schema interno del DAC 0830

Il DAC 0830 ha un ingresso parallelo a 8 bit e un'uscita in corrente. Per ottenere l'uscita in tensione nel range desiderato è necessario inserire un convertitore corrente tensione e un amplificatore.

In Fig.8 si riporta lo schema elettrico del sistema di conversione Digitale/Analogico in grado di fornire in uscita in tensione variabile tra 0 e 10V.

È un convertitore digitale - analogico ad 8 bit basato su una rete di resistenze R-2R e quindi presenta due uscite in corrente.

Il circuito è costruito da 2 registri a 8 bits ciascuno dei quali realizzato da 8 flip - flop D, da una rete di resistenze R-2R e da una logica combinatoria di controllo.

La memorizzazione della parola binaria applicata agli ingressi di ciascun registro avviene portando a livello basso la linea LE (latch-enable). Se  $LE=1$  le uscite Q inseguono gli ingressi D.

Il convertitore presenta le seguenti linee:

$\overline{CS}$  : Chip select (attiva bassa). In combinazione con ILE abilita Wr1.

ILE : Input latch enable (attiva alta).

1  $\overline{WR1}$  : Write 1 (attiva bassa). È usata per caricare i bit del dato di ingresso digitale (DI) nel primo registro. Il caricamento avviene ponendo  $Wr1=0$  se  $ILE=1$  e  $CS=0$ . Il dato rimane memorizzato nel registro (latch) se  $\overline{WR1}=1$ .



## PORTA CENTRONICS

L'interfaccia più usata per la sua semplicità e duttilità è la *Centronics*, un'interfaccia parallela di tipo asincrona usata generalmente per collegare un computer alla stampante o allo scanner. In quelle di ultima generazione il flusso di dati è bidirezionale.

Essa è accessibile all'esterno del computer normalmente attraverso un connettore di tipo D a vaschetta a 25 poli femmina. Sono disponibili 8 linee per dati, 9 per segnali, di ingresso o di uscita, che permettono il colloquio tra la periferica e il computer, e più linee, collegate a massa. Nella tabella è riportato l'elenco completo dei segnali presenti sul connettore, con il relativo numero di pin, e in figura il disegno del connettore stesso.

PIN	NOME	DIREZIONE	INDIRIZZO ED USO
1	$\overline{\text{STROBE}}$	USCITA	890 OUT 890,1 0V 890 OUT 890,0 5V
2	DATA1	USCITA	OUT 888,N Per $0 \leq N \leq 255$
3	DATA2	USCITA	
4	DATA3	USCITA	
5	DATA4	USCITA	
6	DATA5	USCITA	
7	DATA6	USCITA	
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	SELECTED	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		

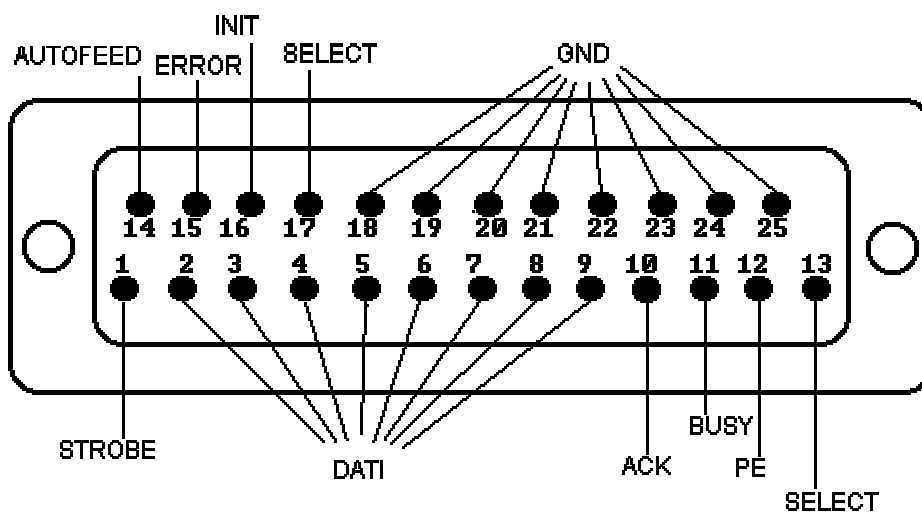


Fig. 10 – Connettore della porta Centronics

L' *interfaccia* consente di acquisire e distribuire dati digitali e impiegando dei convertitori DAC e ADC e pochi componenti di controllo è possibile anche la distribuzione e l'acquisizione di segnali analogici.

Per gestire la Centronics in ambiente VB6 è necessario controllare via software tre registri della porta allocati agli indirizzi: 888, 889, 890.

### **INDIRIZZO 888**

Le linee di uscita DATA8...DATA1, di indirizzo 888 (378 esadecimale), situate tra i pin 9...2, sono memorizzate all'indirizzo 888 :

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

### **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).

<b>LINEE</b>	$\overline{\text{BUSY}}$	<b>ACK</b>	<b>PAPER OUT</b>	<b>SELECTED</b>	<b>ERROR</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>PIN</b>	<b>11</b>	<b>10</b>	<b>12</b>	<b>13</b>	<b>15</b>			

### **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.

<b>indirizzo 890</b>	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
<b>LINEE</b>				<b>IRQE</b>	$\overline{\text{SI}}$	<b>IP</b>	$\overline{\text{AF}}$	$\overline{\text{STB}}$
<b>PIN</b>					17	16	14	1

Dell'indirizzo 890 è molto importante il bit 5 che in base al suo stato può decidere se l'intero indirizzo 888 è impostato in uscita o in ingresso. Se il bit 5, è al livello logico 1 l'indirizzo 888 è impostato come input se invece il livello logico è 0 l'indirizzo 888 è impostato come output. Questo bit è da tenere in considerazione soprattutto in fase di progettazione del software quando, prima di ogni altro procedimento, l'indirizzo 888 deve essere impostato come output o come input.

## SOFTWARE DI COMANDO IN VB6

Nel software realizzato per la comunicazione con il DAC si è tenuto presente che l'indirizzo 888 comprende gli 8 bit paralleli, e il 5 bit dell'890 dichiara, tramite il suo stato, se l'indirizzo 888 è di input(1) o di output(0).

Il software, realizzato con Visual Basic 6, ha la funzione di porre in uscita dalla porta Centronics un numero binario parallelo a 8 bit, che varia da 0 a 255. Poiché VB6 non supporta le operazioni input ed output su Centronics si è utilizzata una DLL per VB6 che è in grado di gestire la parallela anche nei sistemi operativi 2000, NT e XP oltre ai sistemi operativi precedenti come Win98. Tali componenti sono disponibili su internet, nella nostra applicazione si è utilizzato il componente port95nt.exe scaricato gratuitamente dal sito Internet [www.driverlinx.com](http://www.driverlinx.com). Attivando il download si può scaricare il file Windows 95NT PortI/O driver che consente di caricare il file port95nt.exe che installa automaticamente diversi file tra cui:

- DLPortIO.DLL -- Win32 DLL hardware I/O functions
- DLPortIO.BAS -- Visual Basic interface to DLPortIO.DLL
- DLPortIO.SYS -- Kernel mode driver for WinNT

(non necessario per Win95)

Si deve inserire il modulo DLPortIO.BAS e le istruzioni di Input e Output devono avere il seguente formato:

- Value = DPortReadPortUchar(Addr)
- DPortWritePortUchar (Addr), Value

Dove Addr è l'indirizzo della porta, ad esempio 888, e Value è il valore in decimale in lettura o in scrittura.

Dopo aver installato il file port95nt.exe si deve aprire VB6 e inserire il modulo DLPortIO.BAS come mostrato in fig.11.

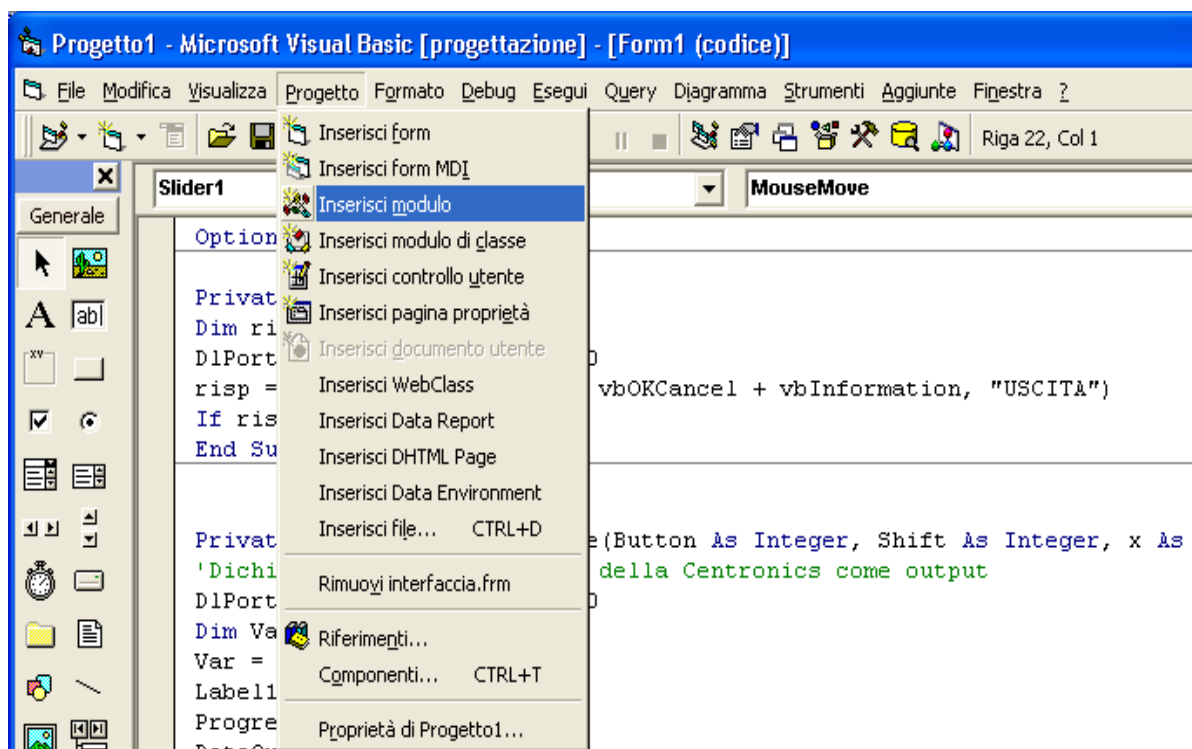
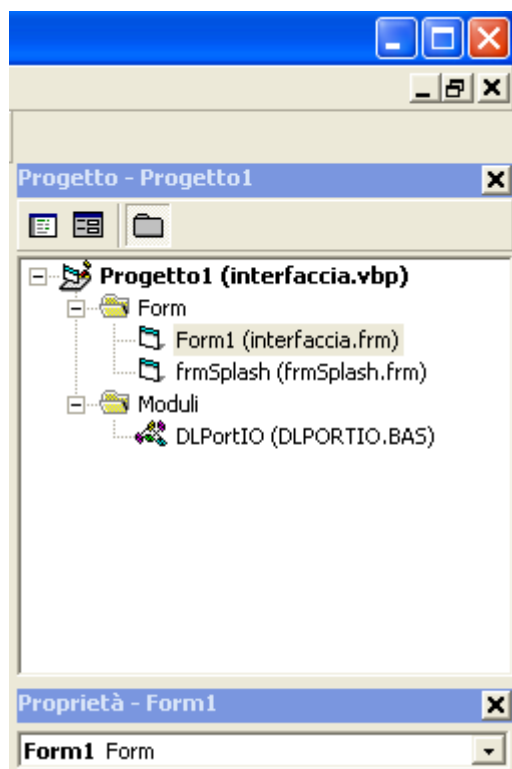


Fig. 11 – Inserimento di un modulo

Dopo aver inserito il modulo, nella schermata di visualizzazione Progetto di VB6 verrà inserita una cartella con il nome Moduli contenente il modulo BAS scelto come indicato in fig. 12.



**Fig. 12 – Finestra Progetto di VB6**

Si riporta il codice associato al modulo DLPortIO.BAS.

```
DIPortReadPortUchar
DIPortReadPortUshort
DIPortReadPortUlong
```

```
DIPortReadPortBufferUchar
DIPortReadPortBufferUshort
DIPortReadPortBufferUlong
```

```
DIPortWritePortUchar
DIPortWritePortUshort
DIPortWritePortUlong
```

```
DIPortWritePortBufferUchar
DIPortWritePortBufferUshort
DIPortWritePortBufferUlong
```

Nel programma è stata utilizzata l'istruzione, DIPortWritePortUchar che consente di scrivere nella porta il valore desiderato.

### Descrizione del codice del programma per il controllo del motore.

Si riporta la schermata del form del progetto. Il software è costituito da due subroutine. Una necessaria per l'uscita dal programma l'altra per mandare il dato variabile da 0-255 all'indirizzo 888 della porta Centronics.

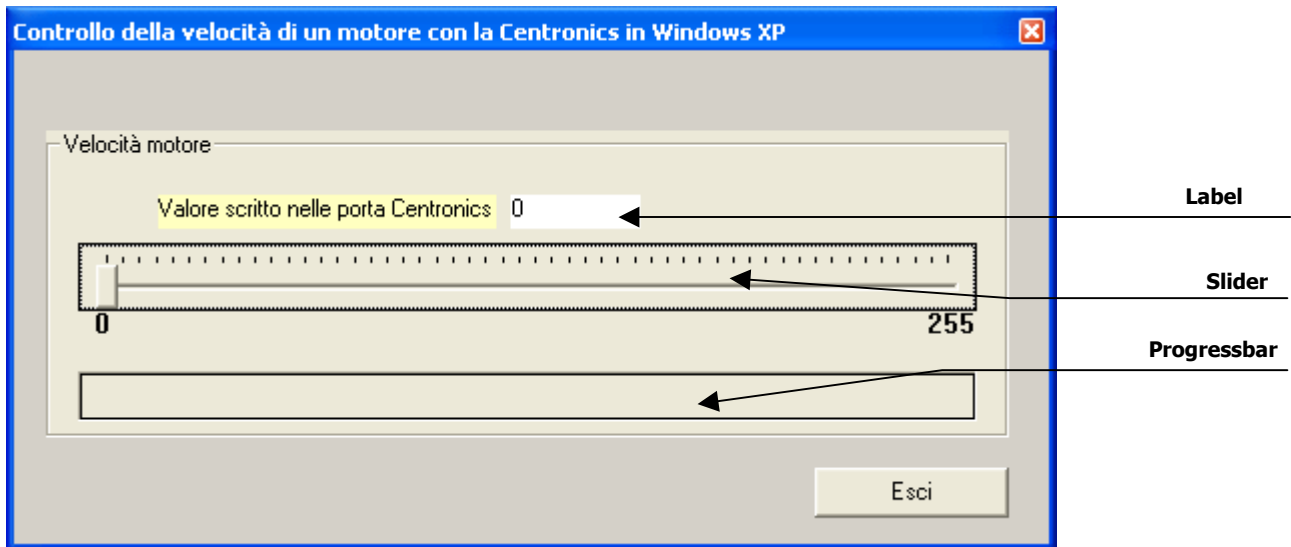


Fig. 13 – Schermata del Programma

#### Private Sub cmdEsci\_Click()

```
Dim risp As Integer
DIPortWritePortUchar (888), 0
risp = MsgBox("Sei sicuro?", vbOKCancel + vbInformation, "USCITA")
If risp = vbOK Then End
End Sub
```

#### Private Sub Slider1\_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)

```
'Dichiara la l'indirizzo 888 della Centronics come output
DIPortWritePortUchar (890), 0
Dim Var, DatoOut As Byte
Var = Slider1.Value
IblDato.Caption = Var
ProgressBar1.Value = Var
DatoOut = Var
'Trasferisce il valore DatoOut all'indirizzo 888 della Centronics
DIPortWritePortUchar (888), DatoOut
End Sub
```

Nel programma sono stati inseriti dei componenti. Per farlo è sufficiente cliccare con il tasto destro sul tool contenente i componenti e selezionare componenti come visualizzato in fig. 14.

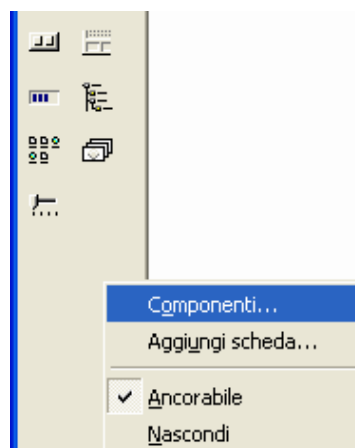
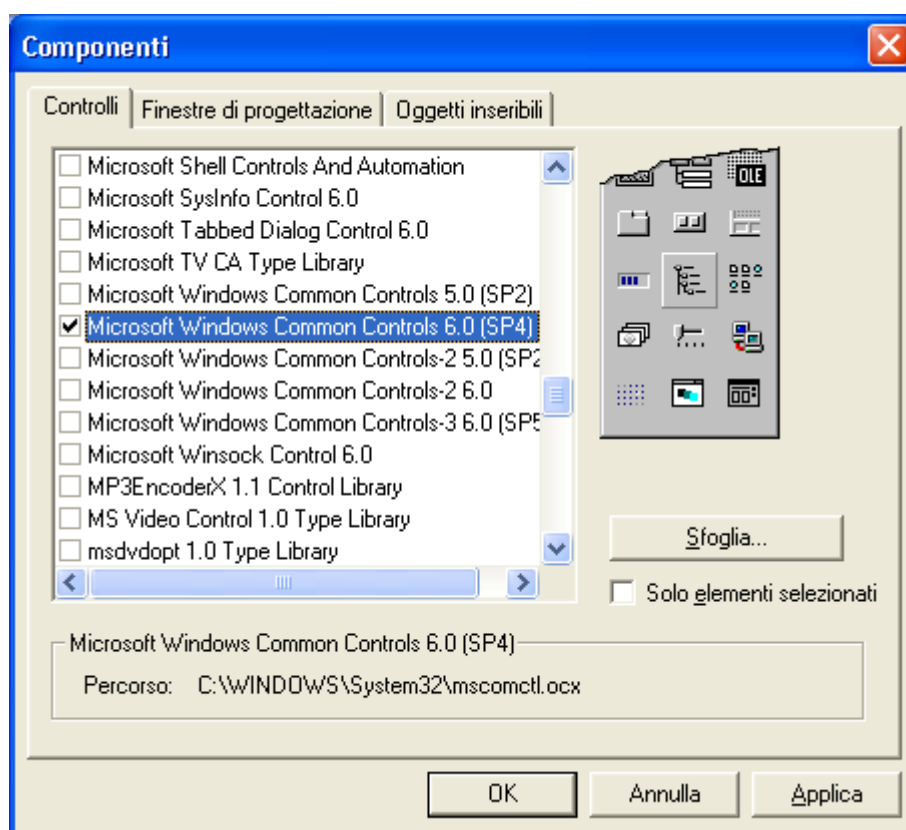


Fig. 14 – Inserimento di un componente aggiuntivo

Ora si può accedere alla lista dei componenti previsti dalla Microsoft come si può vedere in fig. 15.



**Fig. 15 – Lista componenti**

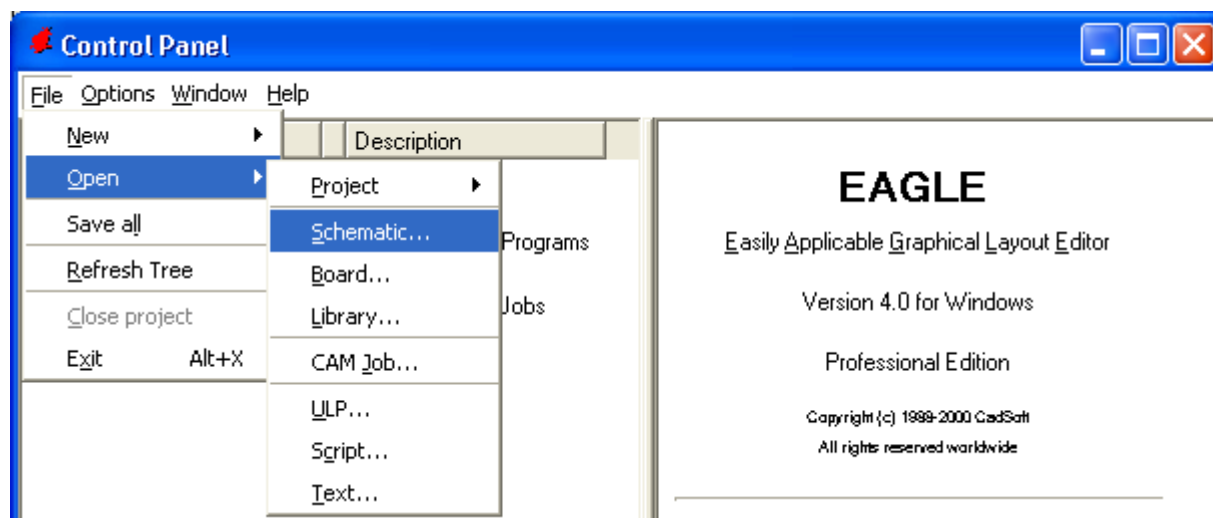
Dopo aver spuntato il componente che si vuole inserire è necessario spingere OK e appariranno sul tool. Il componente inserito è Microsoft Windows Common Control 6 e sono:

1. Lo Slider, che simula un potenziometro lineare di tipo a striscia.
2. Progressbar., che simula una barra di visualizzazione.

## REALIZZAZIONE PCB CON EAGLE

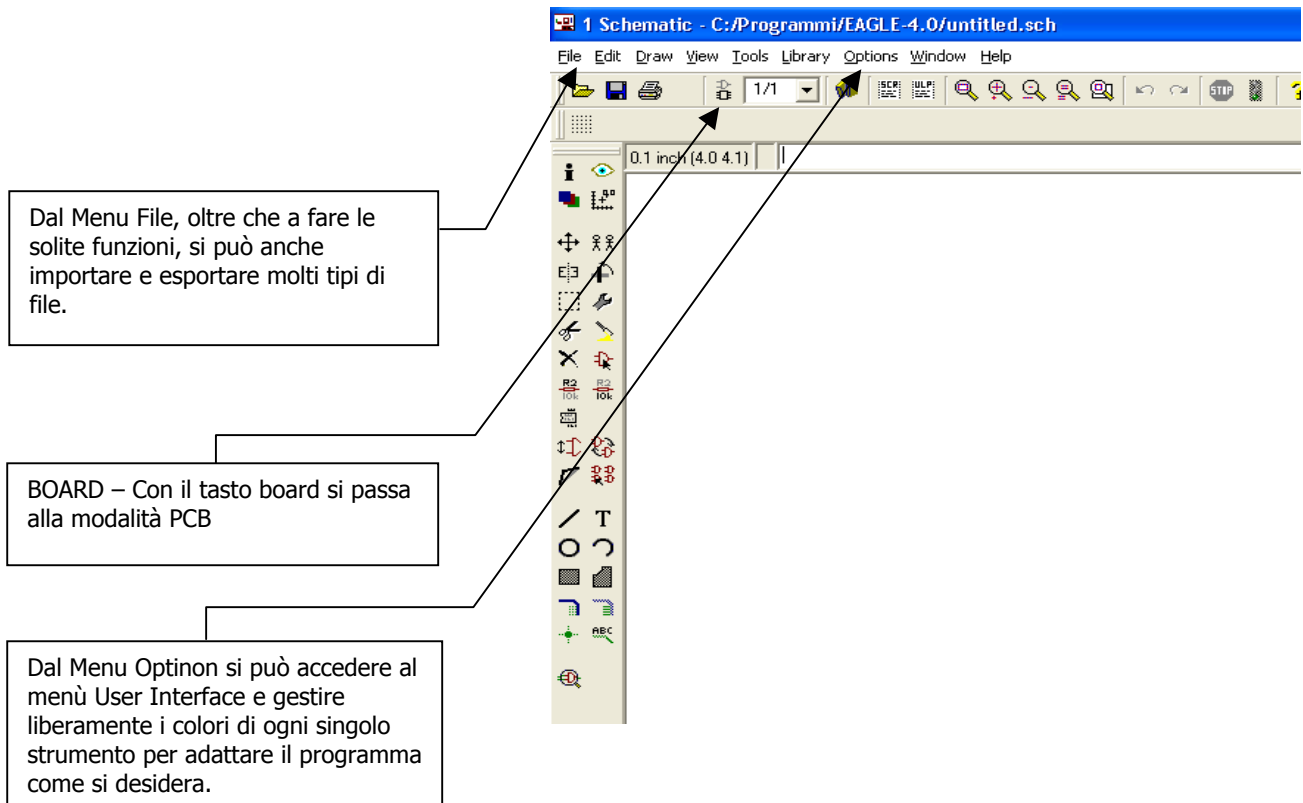
Eagle è un programma per la realizzazione di PCB. Esso è scaricabile direttamente dal sito [www.cadsoft.de](http://www.cadsoft.de) in varie versioni sia per Linux che per Windows.

EAGLE quando si apre presenta una Control Panel, esso è il centro di controllo da dove si possono avviare varie sessioni tra cui quella per la realizzazioni di PCB attraverso il pulsante File – New – Schematic.



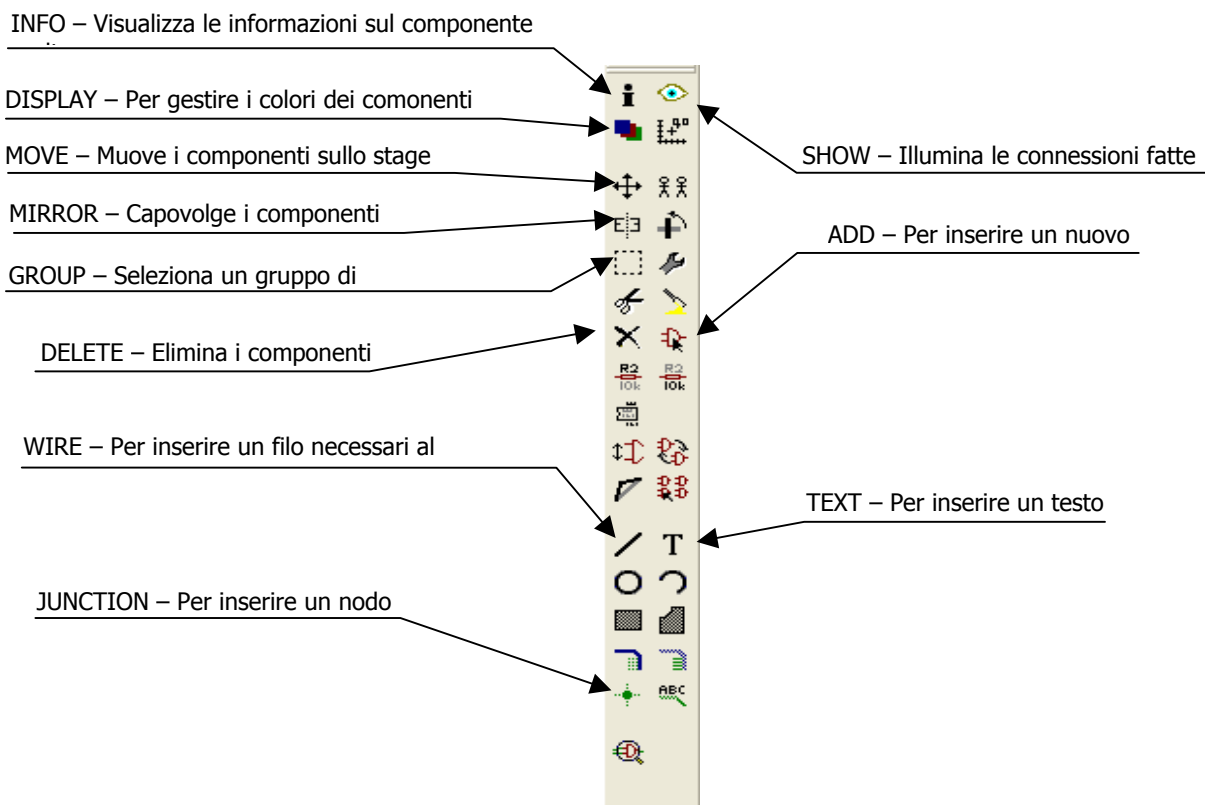
**Fig. 16 – Control Panel di EAGLE**

L'ambiente Schematic presenta varie opzioni e un tool.



**Fig. 17 – Ambiente Schematic**

Nel Tool troveremo tutto quello che necessita per disegnare uno schema elettrico.



**Fig. 18 – Tool Schematic**

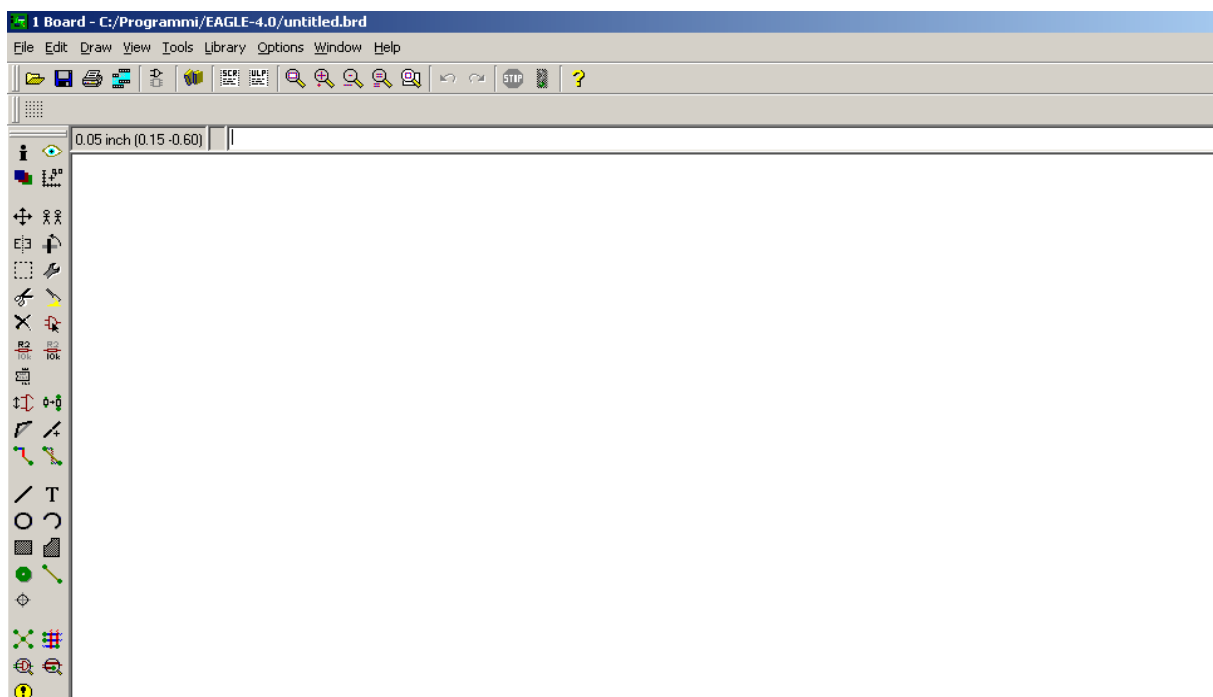
Nell' ambiente schematic con l'ausilio del tool appena descritto, possiamo incominciare a disegnare lo schema elettrico in maniera corretta.

Dopo che lo schema è stato disegnato possiamo passare attraverso il tasto board nell'ambiente PCB, dove il CAD attraverso il nostro schema elettrico ci darà la possibilità di ottenere il PCB.



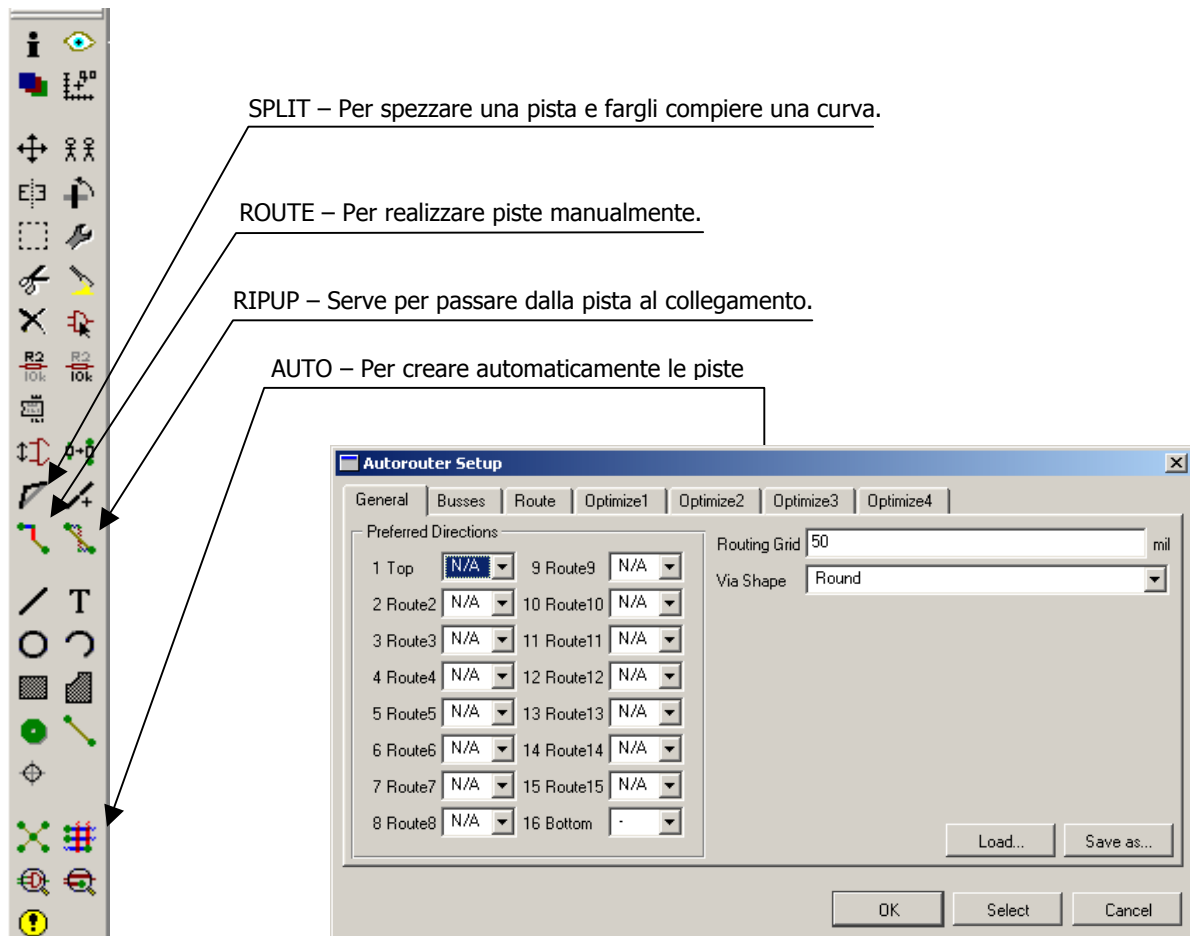
**Fig. 19 – Pulsante Board**

Nell'ambiente Board il tool è formato circa dagli stessi tasti. L'operazione principale, nell'ambiente Board, è quella di posizionare i componenti in una maniera corretta, affinché il cad sia in grado di realizzare un PCB senza ponticelli e senza errori.



**Fig. 20 – Ambiente Board**

I Pulsanti del Tool che si differenziano rispetto a quello dello schematic sono:



**Fig. 21 – Tool Board**

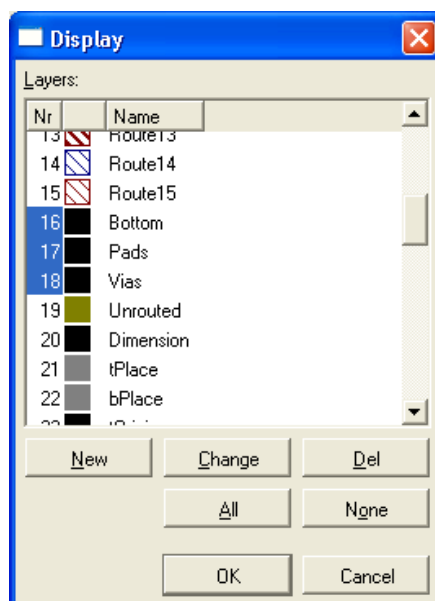
Le impostazioni di Auto sono molto importanti. Per realizzare un PCB a una sola faccia bisogna disattivare il top come si può vedere in Fig.21.

Dopo aver impostato la schermata auto il programma avvierà la ricerca delle piste affinché non le collegherà tutte.

Se non ci dovesse riuscire il collegamento mancante dovrà essere sostituito con un ponticello creandosi da soli la pista.

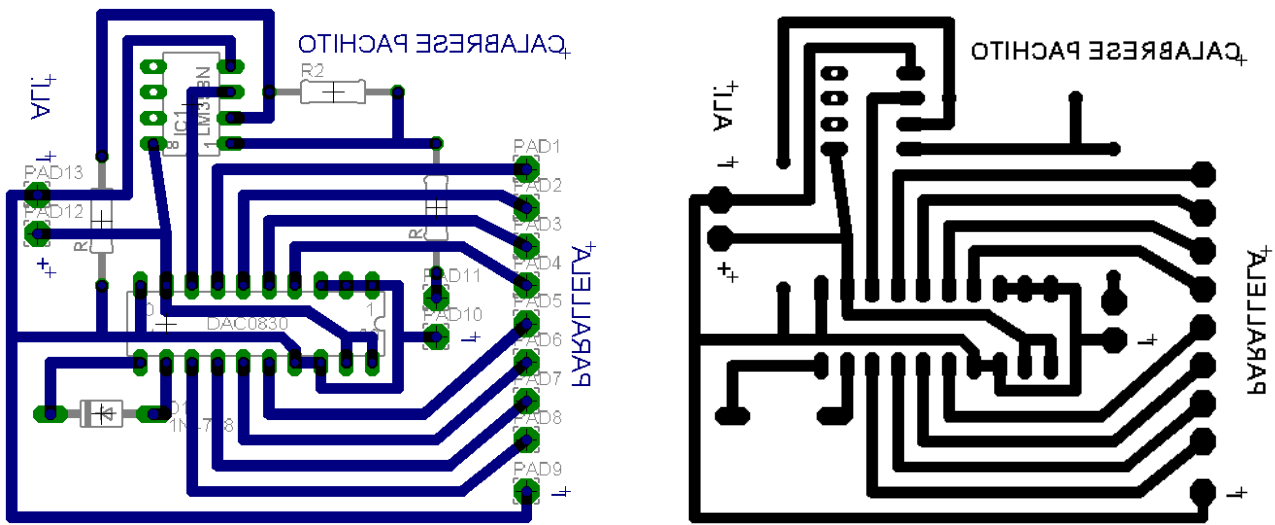
Ora che si è finito il disegno del PCB si passa alla stampa sul lucido.

Per stampare solamente quello che ci interessa come detto precedentemente dobbiamo far uso del menu Display, che ci dà la possibilità di fare in nero solo i pad e le piste, eliminando dal disegno quello che non ci interessa.



**Fig. 22 – Menù Display**

Infatti selezionando e deselezionando quello che si desidera si ha la possibilità di visualizzare o meno alcuni parti del disegno. Nella fig.22 si è selezionati solo i campi Bottom, Pads, Vias che sono in pratica i tre elementi che formano qualsiasi PCB. Il colore attraverso il pulsante Ch'ange si può cambiare a proprio gradimento ma per una stampa nitida (e soprattutto per il Bromografo) è consigliato scegliere il colore nero.



**Fig. 23 – Trasformazione per la stampa sul lucido dopo aver effettuato le modifiche nel menù Display**

Ora che il PCB è stampato sul lucido è sufficiente avere un bromografo una basetta presensibilizzata e procedere alla realizzazione vera e propria.