

ISTITUTO SUPERIORE  
Fossati - Da Passano  
A.S. 2012/2013

## Relazione Tecnica

# *Cablaggio ponte H con integrato L293B*

***Redatori:***

***Bianchi Claudia***

***Gjoni Juela***

***Manfredi Francesca***

***Musso Giada***

***Pelati M. Chiara***

# INDICE

1.0: Componenti utilizzati

1.1 Il ponte ad H

1.2 Il cablaggio

1.3 I transistor

1.4 Il bjt

2.0 Onda quadra

2.1 PWM (pulse width modulation)

2.2 VBB e Programmazione con Arduino relativo al punto 2.1

3.0 Schema integrato L293-B

3.1 Schema integrato

## 1.0 componenti utilizzati

- Integrato di potenza L293-B
- Shield Arduino
- 2 motori in corrente continua da 5V a 20V

## 1.2 Il ponte ad H

Per fare girare il motore in entrambi i versi di rotazione è necessario invertire il segno della corrente che passa all'interno del motore stesso. Per far ciò si usa un circuito chiamato ponte H costituito da 4 interruttori comandati e da 4 diodi di ricircolo.

Il nome deriva dall'assomiglianza del circuito alla lettera maiuscola H, dove il motore costituisce il segmento orizzontale e i 4 transistor i quattro segmenti verticali. In genere i due transistor inferiori sono detti di *sink* in quanto assorbono la corrente proveniente dal motore oppure *low side switch*; i due transistor connessi direttamente alla VCC sono detti *source* oppure *high side switch*.

A seconda di quali transistor sono attivi, abbiamo diversi possibili percorsi per la corrente.

- Un ponte ad H sfrutta 4 transistor, due del tipo NPN e due del tipo PNP collegati come nella figura seguente:

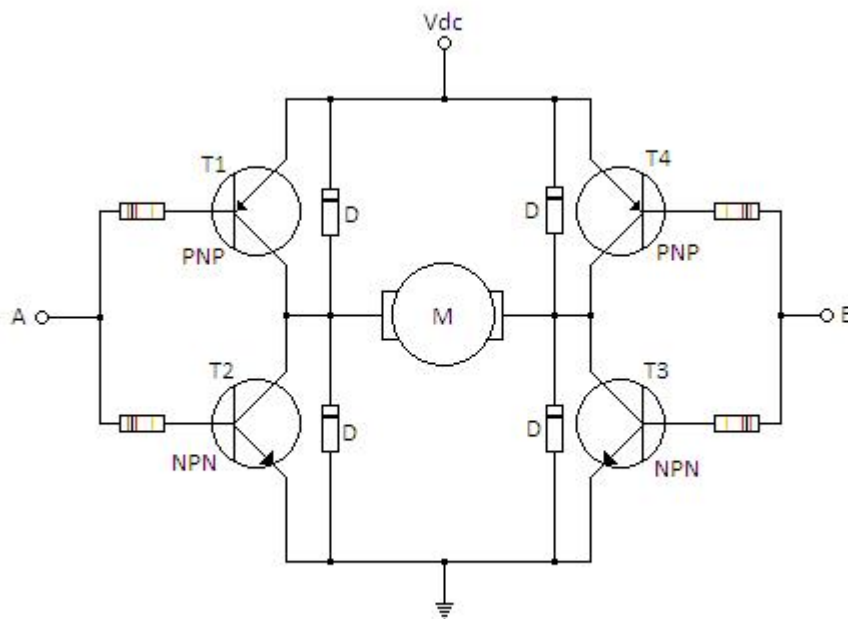
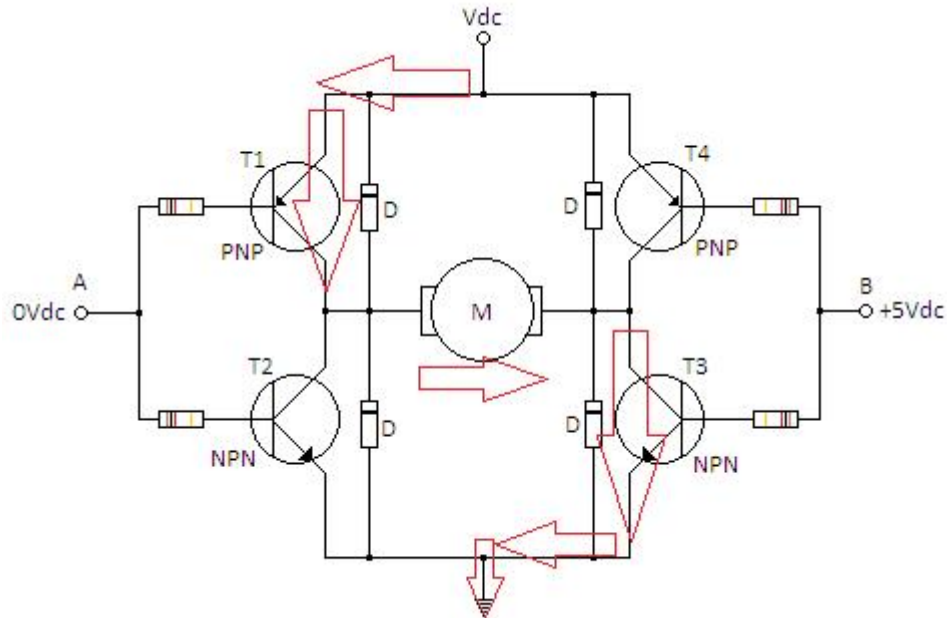


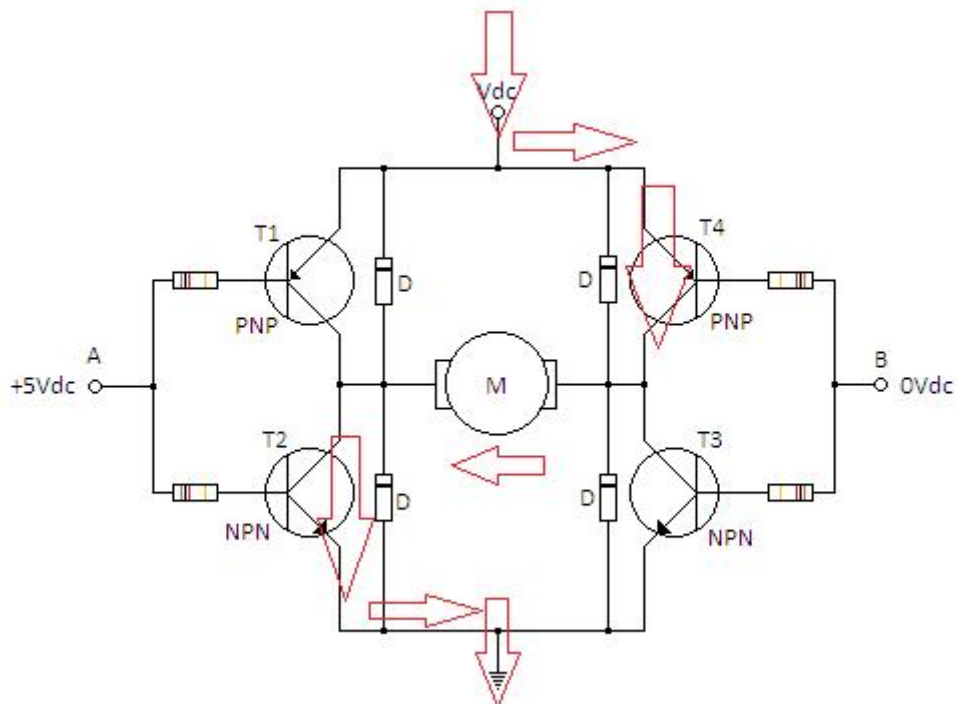
figura 1.a

- Il segnale d'ingresso viene applicato al **pinA** e al **pinB** del ponte H. Quando A è alto e B è basso vanno in conduzione i transistor **T4** e **T2** provocando la rotazione a sinistra del motore.



*Figura 1.b*

Quando sul **pinA** il segnale è basso e sul **pinB** il segnale è alto i transistor che vanno in conduzione sono il **T1** e il **T3**, provocando la rotazione a destra del motore.



*Figura 1.c*

## 1.2 Cablaggio

In telecomunicazioni e informatica il cablaggio è l'insieme dei collegamenti e impianti fisici (cavi, connettori..) che permettono l'interconnessione a livello di rete locale, tipicamente nell'ambito di un edificio o di un gruppo di edifici, di:

- terminali telefonici ovvero telefoni attraverso linee telefoniche dirette verso un centralino telefonico;
- terminali (*host*) di rete sotto forma di una rete di calcolatori interna (*LAN*);

Ognuna di queste infrastrutture di rete locali possiede debite differenze tecniche e tecnologiche di interconnessione. Le caratteristiche elettriche, le lunghezze dei cavi e dei connettori impiegati influenzano le tipologie di reti locali realizzabili. Tra i primi esempi di cablaggio si possono citare le prime reti ethernet che sono dette 10base5, *thick cable*, costituite da grossi cavi coassiali in rame, a cui i calcolatori dovevano essere collegati perforando la guaina esterna fino a raggiungere il connettore esterno.

Questi cablaggi avevano una tipologia a bus.

## 1.3 I transistor

Un transistor è un dispositivo elettronico a tre terminali : *Base*, *Collettore* ed *Emettitore*.

Il transistor ha due modalità di funzionamento : come *amplificatore* e come *interruttore*.

Il funzionamento ( come interruttore) è il seguente :

- Se la corrente entrante nella Base è nulla, anche la corrente entrante nel Collettore e quella uscente dal Emettitore sono nulle. Se consideriamo i due terminali C ed E si tratta quindi di un comportamento simile ad un interruttore aperto.
- Se la corrente entrante nella Base è "sufficiente", possiamo avere una corrente entrante dal Collettore ed uscente dall'Emettitore. La tensione tra C ed E è molto bassa, (spesso approssimata a 0V). Se

consideriamo i due terminali C ed E si tratta quindi di un comportamento simile ad un interruttore chiuso.

## 1.4 Il bjt

Il transistor, opportunamente polarizzato, può essere utilizzato come un interruttore che può essere aperto o chiuso regolando la corrente di base

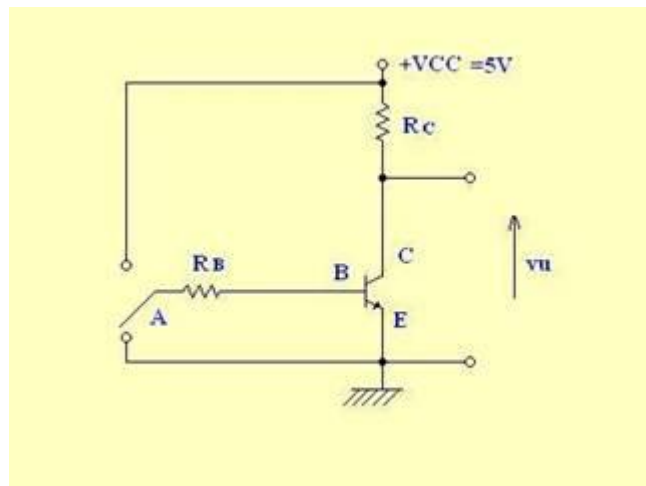


Figura 2.a

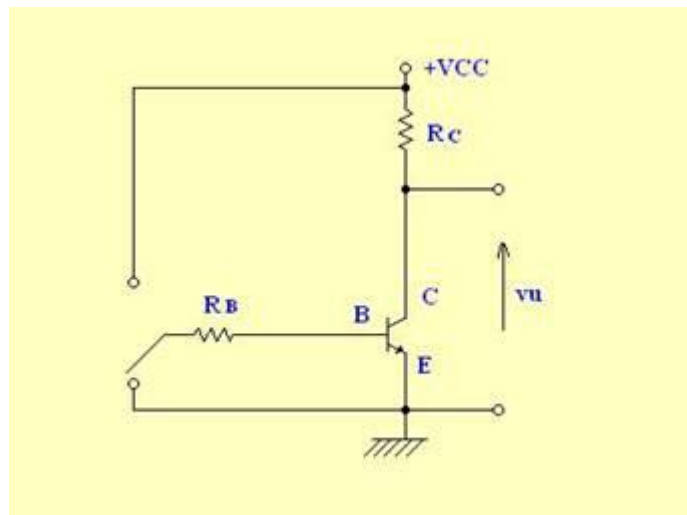


Figura 2.b

Quando l'interruttore si trova verso il basso la tensione  $V_{BE} = 0$ ;  $I_B = 0$ ;  $I_C = 0$ .

Il transistor è interdetto, non conduce, e si comporta da circuito aperto.

La tensione di uscita sul collettore, raggiunge il valore massimo  $v_u = V_{CC}$ .

Quando invece spostiamo l'interruttore verso l'alto, la base del transistor è polarizzata direttamente, e il transistor va in saturazione, la  $I_C$  raggiunge il massimo valore, il transistor si comporta da circuito chiuso, e la tensione di uscita assume il valore  $v_u = 0$ .

A	Y
0	0
1	1

## 2.0 Onda quadra

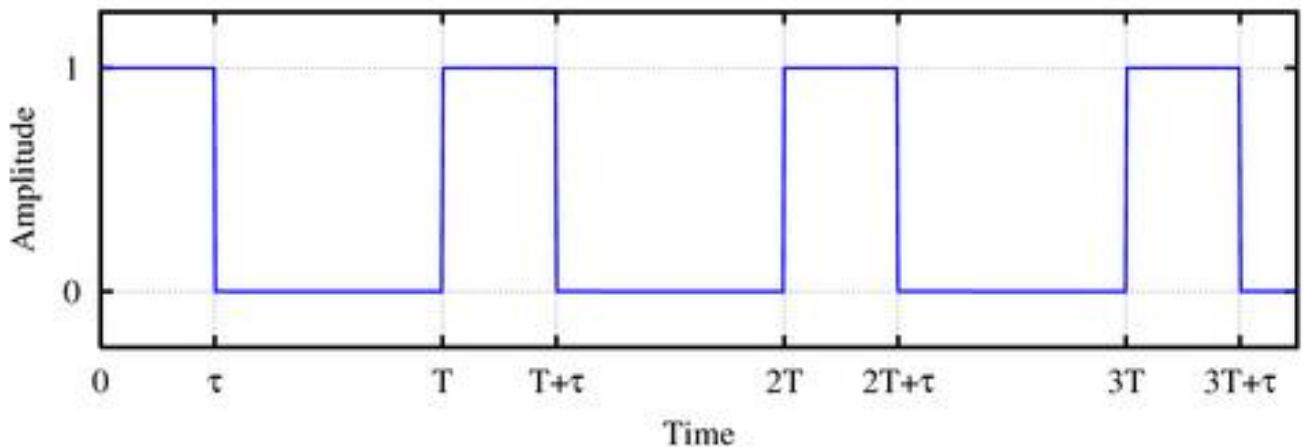


Figura 3.a: onda quadra con Duty cycle 30% circa;

In elettronica l'**onda quadra** è un segnale composto da un'alternanza regolare di due valori, che sono equivalenti al segnale elettrico utilizzato nei circuiti digitali. I due livelli possono essere messi in corrispondenza con i due stati logici 0 e 1.

Nel caso di un segnale elettrico, un'onda quadra è composta da due livelli di tensione. Esempio tipico di onde quadre sono i segnali digitali, nei quali il tempo di transizione dal livello alto al livello basso, detto "fronte di salita", e da quello alto a quello basso, detto "fronte di discesa", è molto breve, ed è determinato dal grado di velocità di commutazione del transistor.

### Duty cycle:

E' la frazione di tempo che un entità passa in uno stato attivo in proporzione al tempo totale considerato.

Spesso il *duty cycle* è indicato sotto forma di percentuale ( $D\%$ ): per ottenere la percentuale basta moltiplicare per 100 il risultato del rapporto  $\tau/T$ . La percentuale esprime più chiaramente il quantitativo di segnale alto (se  $D=0,4$ ,  $D\% = 40\%$ , quindi significa che per il 40% del periodo totale il segnale è a livello alto). In particolare, se  $D=0,5$  ( $D\%=50\%$ ) significa che

per metà del periodo totale il segnale è alto, per l'altra metà è basso: siamo quindi in presenza di un'onda quadra.

## 2.1 PWM (Pulse width Modulation)

E' un tipo di modulazione digitale che permette di ottenere una tensione media variabile dipendente dal rapporto tra la durata dell'impulso positivo rispetto a quello 0.

Il compito principale del PWM è quello di risparmiare energia.

L'applicazione del PWM nel progetto della nostra compagna è legata all'accensione graduale di un led ( intensità massima 255).

Tenendo premuto un tasto inserito nella breadbord, controllato dal pin2 e il pin11 (Questo tasto è abilitato al PWM) dell'Arduino, il led aumenta la sua intensità gradualmente; al contrario, smettendo di premere il tasto, il led si spegne a poco a poco.

## 2.2 VBB e Programmazione con Arduino relativo al punto 2.1

```
volatile boolean Tasto=false;
unsigned int led, Pin2, Pin13, Pin11, IntensitaLuminosa; //definisce il
pin di ingresso digitale e le variabili intere senza segno
```

```
void setup ()
{
  Pin2=2; // corrisponde al pin che verrà usato per leggere lo stato del
pulsante
  Pin11=11; // corrisponde ad uno dei pin PWM usato per regolare in
uscita la tensione del led
  Pin13=13; //corrisponde al pin a cui è sempre collegato il led di
arduino sulla piastra madre
  led=Pin13;
  IntensitaLuminosa=0;
```

```

    pinMode(led, OUTPUT); //inizializza la variabile led
    corrispondente al pin13 come output
    pinMode(Pin2, INPUT); //inizializza il pin2 come input

}

void loop ()
{
    leggiTasto();//chiamata della funzione
    if (Tasto==HIGH)
    {
        Lampeggia();
        analogWrite(Pin11, IntensitaLuminosa); //pin11=piedino su cui si
        invia il segnale ed è un piedino PWM(modulazione di larghezza di
        impulso) intensitaluminosa= varia da 0 a 255 capacità luminosa del
        led.
        Aumenta();
        Ritardo(20);
    }
    else
    {
        Spegni();
        Diminuisci();
        analogWrite(Pin11, IntensitaLuminosa);
        Ritardo(50);
    }
}

void leggiTasto()
{
    Tasto=digitalRead(Pin2); //legge il valore del pin2
}

void Accendi()
{
    digitalWrite(led,HIGH);
}

void Spegni()

```

```

    {
        digitalWrite(led,LOW);
    }
void Aumenta()
{
    if(IntensitaLuminosa<255) IntensitaLuminosa++;
}
void Diminuisci()
{
    if(IntensitaLuminosa>0) IntensitaLuminosa--;
}
void Ritardo(int Valore)
{
    delay(Valore);
}
void Lampeggia()
{
    Accendi();
    Ritardo(1);
    Spegni();
    Ritardo(50);

}

```

### 3.0 VBB e Programma con Arduino

```

int pwmpin3=3;
int pwmpin9=9;
int c=4;
int d=5;
int e=2;
int f=1;

void setup()
{
    //put your setup code here, to run once:
    pinMode (4,OUTPUT);
    pinMode(5,OUTPUT);

```

```
pinMode(1,OUTPUT);
digitalWrite (4,HIGH);
digitalWrite (5,LOW);
digitalWrite (1,HIGH);
digitalWrite (2,LOW);
analogWrite(3,255);
analogWrite(9,255);
}
```

```
Void loop()
```

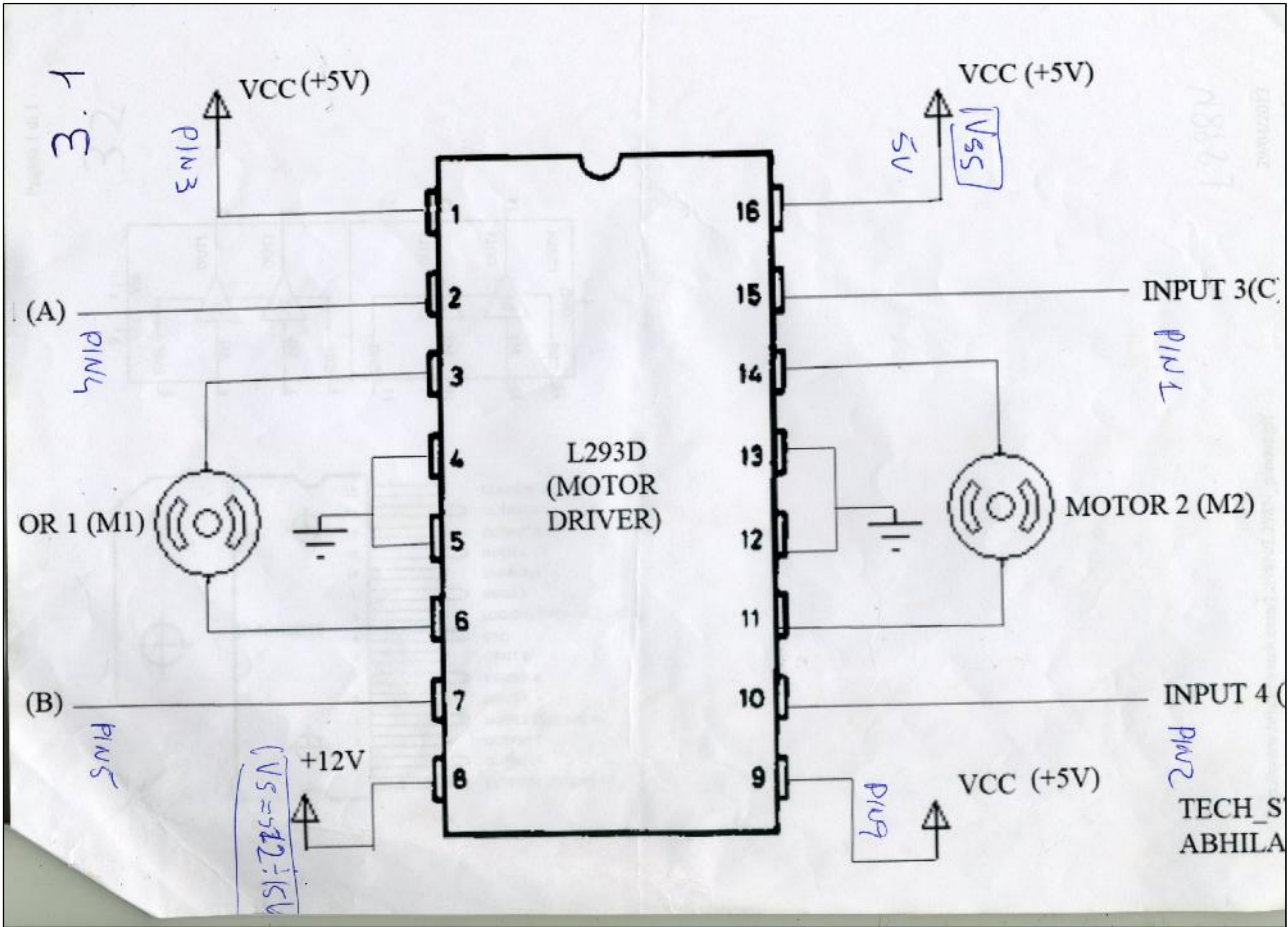
```
{
```

```
//put your main code here,to run repeatedly:
```

```
delay(15000);
digitalWrite(4,LOW);
delay(3000);
digitalWrite(4,HIGH);
delay(7000);
digitalWrite(4,LOW);
delay(3000);
digitalWrite(4,HIGH);
delay(15000);
digitalWrite(4,LOW);
delay(3000);
digitalWrite(4,HIGH);
delay(7000);
digitalWrite(4,LOW);
delay(3000);
digitalWrite(4,HIGH);
```

```
}
```

### 3.0 Schema integrato L293D



### 3.1 Schema integrato L-298

